uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES**

uOttawa

L'Université canadienne
Canada's university

**FACULTY OF GRADUATE AND POSDOCTORAL STUDIES**

Ismaeel Al Ridhawi

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Design and Implementation of a Location Aware User Tracking System

TITRE DE LA THÈSE / TITLE OF THESIS

A. Karmouchl

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

B. Esfandiaril

A. Nayakl

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

# Design and Implementation of a Location-Aware User Tracking System

**Ismaeel Al Ridhawi**

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MASc degree in Electrical Engineering

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario

*Your file Votre référence*
*ISBN:* 978-0-494-61304-7
*Our file Notre référence*
*ISBN:* 978-0-494-61304-7

# Table of Contents

# List of Figures

# List of Tables

# Abstract

Communication devices are establishing themselves as ubiquitous features of users' daily life. This advancement is linked to tremendous growth in the sophistication of mobile-aware applications. Such applications need access to information about the users' and other objects' physical locations, a requirement known as location-awareness.

Existing location-aware applications and systems are typically restricted to a particular type of positioning technology, while modern location-aware services and applications use context and prediction methods to adapt to the needs of users and changes in the environment. The use of a single sensing technology alone provides a less accurate estimation of a user's current location.

In this paper, we introduce an ontology-based location tracking system. It makes use of multiple sensing technologies and includes a prediction method for identifying a user's current location and predicted future location. Our system architecture better serves the client by using location and context information and is, therefore, an important step forward in predictive tracking.

# Acknowledgements

I must first express my deepest gratitude to my supervisor, Professor Ahmed Karmouch; his valuable guidance, constructive advice, academic support and encouragement have been vital to the progress of my research and completion of this thesis.

I would also like to thank Yousif Al Ridhawi, Stenio Ferandes, Moyad Al-Oqily, Nancy Samaan, Ibrahim Al-Oqily, Majdi Rawashdeh, and other members of the Intelligence for Mobile Autonomic and Cognitive Networks Laboratory for their help and support during my research. Their companionship and friendship eased many tasks during times of difficulty.

Last, but definitely not least, I should express my boundless gratitude and love to my parents whose consistent encouragement, support and endless love have taken me through all hardships incurred during my studies and research journey. To you I dedicate this thesis, for teaching me that nothing in life is impossible.

# Acronyms

| | |
|---|---|
| AP | Access Point |
| CC/PP | Composite Capabilities / Preferences Profiles |
| COMANTO | COntext MAnagement oNTOlogy |
| CONON | OWL Encoded Context Ontology |
| CSCP | Comprehensive Structural Context Profiles |
| FOAF | Friend of a Friend |
| GPS | Global Positioning System |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| LBS | Location Based Services |
| LDAP | Lightweight Directory Access Protocol |
| MAC | Media Access Control |
| ORL | Olivetti and Oracle Research Laboratory |
| OWL | Web Ontology Language |
| PDA | Personal Digital Assistant |
| RDF | Resource Description Framework |
| RFID | Radio Frequency Identification |
| RSS | Received Signal Strength |
| SCAPE | Sentient Communication Architecture for Pervasive Environment |
| SCE | Semantic Context Entity |
| SOCAM | Service-Oriented Context-Aware Middleware |
| WLAN | Wireless Local Area Network |
| XML | Extensible Markup Language |

# Chapter 1

## Introduction

### 1.1 Motivation

In the past, personal computers have been restricted in their ability to adapt to the needs of users and to changes that occur in the surrounding environment. Devices of this type were unaware of the environment within which they operated and did not have the ability to react or adapt to changes that occurred around them. Users must manually choose adequate applications or select their current location in order for the system to perform the personalized user settings. Applications were constructed to compute simple tasks and to meet certain requirements with no changes in the environment surrounding the devices and their users. But with the introduction of mobile systems, research has been aiming to solve user concerns and requirements with the least amount of user intervention. Ambient Intelligence [1] today plays a role in everyday life where devices work together to support users in carrying everyday life activities and tasks. But it is clear to us today that these portable computers and mobile devices we carry everywhere we go provide few services in response to the location we are in. Users want all their work information downloaded to their laptop once they arrive to the office; or know of nearby printers inside a certain building of the campus beforehand. To provide these services, it is essential to develop a location-aware system.

Location-awareness serves a major role in enhancing ubiquitous environments [2]. It delivers services based on personal and general context to the mobile user according to the location they are in. Many applications developed today, such as medical services, home applications, office applications, and university services, favor using location and context information. Fast, accurate, and cheap location tracking techniques is a main issue in current research attempts. The importance of mobility location techniques can be seen very clearly today as we begin a new era of dependence on these types of devices to get to places with ease and comfort in our daily lives.

Location-based services are very wide in scope, and every device can accommodate the use of location-aware applications, some of the categories of these services are:

- Location tracking applications, where location information is sent to remote parties such as servers, to take control of the tracking.
- Use of location information for communication decisions, where different locations may require different communication behaviors. For example, a caller's location may affect communication behavior. If the caller is in a place requiring silence, such as a library, the caller's location agent may only enable text messaging.
- Location changes can trigger agents to take action according to that change. For example, when a user gets close to his office, a location notification is detected and the lights and computers in the office are turned on. Another example, if a child leaves the house area, parents will be notified via cellphone.

- Location-aware systems can be used in resource discovery, to help find nearby resources. For example, a user may be interested in nearby printers; therefore, those resources will be displayed on the guidance map on his/her Smartphone.
- A location can be treated as a communication entity. This location can be used to represent a certain set of people. For example, when a device is used to send a message to location "CBY B502," this message will be broadcasted to all the people in that room.

With the current advances in the field of wireless technology, it is not only essential to locate a mobile user in real-time, but also to predict future locations a user will be heading is a key component in assisting many services. Mobility prediction will be a major source of assistance in handoff management, resource reservation, and service pre-configuration, achieving a seamless transfer or delivery in services. Fast and accurate mobility prediction techniques have become one of the main topics in current research efforts. The importance of mobility prediction techniques can be seen at both the network and service levels.

Location-aware computing refers to a system which has the ability to sense and react to dynamic environments and activities. Recent trends of mobile technologies have highlighted the central role of location-awareness in mobile applications and services. The ability to conceive the status of the internal or external environment and to use this information to improve a system's functionalities in location tracking and prediction of a mobile user will result in a faster and much more accurate resolution.

## 1.2 Definition of Location and Context

An understanding and a description of the term "location" is required first; this description will improve the result in a location-aware system. According to Webster's [3] dictionary definition, location is "a position or site occupied or available for occupancy or marked by some distinguishing feature." Location can be described in three general ways: Geospatial addresses, civil addresses, and location attributes. Geospatial coordinates describe a physical location using longitude, latitude, and altitude values. These coordinates are aimed at describing a physical location for outdoor location tracking uniquely. Geospatial coordinates can be used for location tracking by using GPS receivers (discussed later).

Civil addresses describe outdoor and indoor locations. Information provided can be a postal address of a specific building, when describing an outdoor location, or a specific room, when stating an indoor location. Civil addresses can also be used for location tracking (discussed later). Civil addresses in comparison to geospatial coordinates are easier to understand by end users but are also less accurate and more awkward.

Location attributes provide information such as the type and privacy status, describing factors that may affect communication behaviors of a certain location. For example, the type of place or the number of people in a certain location has an effect on the location. Location attributes can be applied to both indoor and outdoor locations and are used to make communication decisions such as rejecting certain e-mails when the user is in a specific location. Different types of locations require different location detection

technologies. Having defined "location," it is now easy for us to study and research "Location-Aware User Tracking Systems."

Based on the above definition and explanation of the term "location," a user roaming within one of the environments described may be provided with a number of applications and services that not only depend on the current location but also on utilizing the vicinity's available context information to present users with the highest level of adaptability. Location-awareness allows the use of different kinds of context, including current activities, actions, or services needed. Location detection could be achieved through GPS systems, through the signal strength of mobile devices carried by users, or by the use of the many available technologies which will be explained later.

Therefore, we must know the meaning of "context" in order to understand how context information can improve location-aware systems. Due to the generality of this definition, it is important to apply the right definition to the field we are discussing, and since our field is about tracking, we have found that Anind Dey has defined it within our context. His definition states that "context is any information that can be used to characterize the situation of any entity. An Entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves. [4]"

## 1.3 Thesis Objective

The goal of this thesis is to build a system that enables the user to interact with the surrounding based on his or her location. Support for location-awareness in a system must meet the following objectives:

**Context Model:** The model should provide an expressive, flexible, and efficient representation for all sorts of physical and logical measurable characteristics such as object, time, organization, activity and most importantly, location. Essential knowledge of spatial properties of locations should be represented. The model must be independent of application domain and location sensor technology. A number of markup languages have been adopted by ontology applications for the purpose of context modeling. We will be using Web Ontology Language (OWL) [5] to model context within our Location-aware system. OWL describes modeled data by using sets of classes and properties, and provides the flexibility and extensibility necessary within pervasive environments.

**Technology:** The architecture for location-awareness must be open to heterogeneous sensor systems. The most common method of location tracking is by using the Global Positioning System (GPS). The drawback to this technology is its inability to be used indoors. Therefore, other methods must be taken into consideration to solve this issue. Using a combination of IEEE 802.11 WLAN and RFID tagging technologies for position tracking of mobile users allows for services to be deployed at one's convenience, both indoors and outdoors.

**Location Prediction:** Once the current location of a user has been identified, it is required that the system identify the location a user will arrive at in the future. The prediction method must be a mathematical probability theory of evidence that combines pieces of evidence to reach decisions in situations of uncertainty. Therefore, user location prediction is performed using the Dempster-Shafer theory of evidence[18].

## 1.4 Thesis Contributions

We need to achieve the goal of a Location-Aware Predicting System that provides services based on the location and environment a user occupies. Those services differ from one location to another. Some of the services may require a longer time to initialize. Therefore, it is required to predict the next location of the user. Different technologies have different capabilities and assumptions and provide assorted levels of location accuracy. No single location sensing technology has emerged as the best in all kinds of environments. Nonetheless, complexity and price play a major role in deciding which technology is best for tracking purposes. Since our environment is based on indoor locations, the technologies that will be incorporated into our system will involve the use of existing WIFI access points and RFID readers and their correspondent active tags. Our system solution enables the separation of applications from the sensing technologies, making it possible to extend the infrastructure with new location technologies, as they become available, without any changes to existing services and applications.

Our location prediction method will locate the user's next destination using knowledge of user's preferences, goals, and other contexts. The availability of user movement histories from our location detecting method will enhance the prediction method which involves the use of the Dempster-Shafer theory. The successful approach in dealing with problems of combining different bodies of evidence to reach decisions in situations with a high degree of uncertainty has attracted considerable attention to this mathematical theory.

Contextual information concerning location, devices, and users in current location detection is used, and also contextual information concerning the environment, users'

interests, goals, tasks, schedule constraints, and location history, is used in future location prediction. Therefore, a unified method of modeling and representation of context is required in this case. Ontologies represent concepts and relationships by employing a computer-usable data structure while sharing a common understanding of the domains in which the modeled context is used. Therefore, we will be using OWL to interpret context information present within the ontology and to infer new context from existing context and relationships. Context modeling will be achieved with help of Protégé [27] ontology editor and knowledge base framework.

The location-aware system provided in this thesis can be used for ontology-based negotiations [6], seamless video handoff [7], mobility prediction [8], displaying a map of the surrounding area to guide the user inside a building, or printing a document on the nearest printer. This interaction with surroundings varies as the location changes. For example, when the user is in the office, the mobile device will receive data concerning work only, but while the user is in the cafeteria, they will receive data concerning friends and entertainment. The claim of proposing a unique architecture is based on the following considerations:

**Use of Multiple Location Sensing Techniques**: Our research in this area explores two specific technologies to location sensing, WLAN and RFID. The first approach consists of discovering the surrounding access points and measuring the signal strength beacons emitted by each access point to the wireless mobile device. The measured signal strength is then mapped onto a two-dimensional map. The second approach improves the location accuracy by using active RFID tags, where signal strength of beacons emitted by a RFID reader are measured and mapped onto the two-dimensional map.

**Involve the Use of Next Location Prediction with Current Location estimation:** Location Prediction is integrated within our architecture to enhance the system into taking actions before the user arrives at his or her destination. Those actions involve both the network level (e.g. network handover management [9]) and the service level (e.g. ontology-based negotiations, seamless video handoff).

**Ontology:** To fulfill the requirements of the architecture, a repository is needed to store context related to locations, users, environments, surroundings, and so on. To achieve the easy communication between detecting and predicting location, the entities within location-aware systems require a common understanding of how key concepts and knowledge is represented. Therefore, we have based our modeling of acquired context on ontologies.

Ontology is a representation of concepts within a domain and the relationships between those concepts. Ontologies are used to reason about the properties of that domain and can also be used to define the domain. An important feature of ontologies is their use in sharing domain information between different entities. They are widely used today in many system architectures due to the simplicity of many of the applications used to model ontologies.

**Architecture:** Since our system must fulfill multiple target requirements, it is desirable to compose the architecture from a number of independent building blocks that can be re-arranged to meet different sets of target requirements.

## 1.5 Thesis Outline

The remainder of this thesis report is structured as follows:

In Chapter 2, we present background information and related work that has motivated our design of a new ontology-based location-aware user tracking system architecture. This chapter presents the different user tracking models that are currently used, as well as some of the location-aware system architectures and middleware available. We also present their advantages and disadvantages. The chapter will also present different context models that are currently used, along with their advantages and disadvantages.

Chapter 3 presents the ontology. We begin by describing the basic classes of the ontology shared by all entities present within a location-aware system, then move on to the details and the wide variety of concepts from many domains and fields of knowledge as well as their relationships to each other.

In Chapter 4, we present our Location-aware system architecture. The architecture has been designed to overcome some of the limitations found in earlier Location-aware systems and middleware. We discuss the classes of the five architecture components: Repository, Initialization and Filtering Center, Location Tracking Center, Location Conflict Resolver Center, and Location Prediction Center. We also illustrate the importance of using ontologies and discuss the details of the negotiations involved between the components. Then we describe in detail the process of predicting a future location using the Dempster-Shafer theory. We will first give a review of the Dempster-Shafer Theory, then discuss Evidence Combination in detail. We will see how to apply the theory to predict future locations of a user to help better track the user.

In Chapter 5, we present our detailed implementation of a prototype system illustrating the validity of our location-aware system's architecture. We discuss the implementation of each of the five centers in our architecture. We will then discuss results of the evaluation conducted on the system.

In Chapter 6 we conclude this thesis. We summarize our contributions and the lessons learned from our ontology-based location-aware system architecture. We also discuss our future research plans and potential enhancements to our current design.

# Chapter 2

# Background

## 2.1 Objective

The growing usage of mobile phones, PDAs, and laptops and the wide availability of wireless LAN technologies have given researchers a field of interest to develop new location-aware systems that provide users with their needs on the spot. Many of these research efforts involve the use of RFID either to identify the user or enhance the identity of the location the user is in. Wireless Local Area Networks are used to identify the location of a user depending on the signal strength emitted from the access point available. This chapter will describe some of the previous and current work conducted by researchers in the location and context awareness field. The chapter will also give an overview of some of the background information used in this thesis. It will also demonstrate the different context modeling approaches recently used by engineers.

## 2.2 User Tracking Models

Location-awareness in mobile environments is an important research area with a number of approaches. Work in the area makes use of different technologies to locate and track an object, including infrared, radio frequency, ultrasound, magnetic fields, and cellular systems. The following presents some of the work conducted by reaserchers.

### 2.2.1 RADAR

RADAR [10] is a software-only system that uses the wireless local area network to track and locate mobile users. The goal of the system is to enable the user to interact with the surroundings in a building, such as printing a document on the nearest printer or guiding a user to a specific location in the building. The system is designed to provide an 'overlapping coverage' of the access points in the area of interest inside the building.

RADAR uses the signal strength of a beacon emitted from an access point to calculate the location of a mobile user. As the mobile user walks away from the access point, the signal strength will decrease. By using this concept RADAR determines the location of the user by building a database.

The system provides accurate location information only when at least three access points, close enough together that their coverage areas significantly overlap, are deployed.

### 2.2.2 CampusSpace

Using the combination of WLAN and RFID technologies for position tracking of mobile users will allow a system to be developed for location-awareness. A mobile user can interact on the move by sustaining his/her context, such as continuing his/her learning as if he/she is in a university-like setting. In order to create such a system, developers are concerned with the issues of where a mobile user works, where the user is using or going to use such services, and where the WLAN and passive RFID tagging technologies are available.

To have a context-awareness location tracking system, information gathered about the mobile user requires that the user is carrying a mobile device such as a laptop or PDA. The position of the mobile user is determined according to the same process that was described before, which is according to the signal strength of a beacon being emitted from an access point, using a radio map. A mobile user will be located according to the position of passive RFID tagged items around a room. The mobile user is equipped with an RFID reader integrated into the mobile station.

A central server must exist to collect location information from the mobile stations and access points. The location server provides the required context, such as a website that will allow the user to access the internet or email.

Those issues were taken into consideration by Ferscha et al. [11] in their architecture. The system has several drawbacks: (a) it scales poorly due to the limited range of passive RFID tagging, and (b) the RFID tagging involves significant installation and maintenance costs.

### 2.2.3 Software Agents

Giorgini et al. [12] developed an architecture that involves the use of software agents. The reason software agents are being used today is because they fulfill the given task without the extra user involvement. For example, in a real world environment such as a shopping mall where the RFID technology is put into use, there would exist an RFID reader for each section of the mall. The RFID reader with the help of software agents are responsible for each section and signs the shoppers in and out depending on if they are in that section or not. A central server must exist to collect location information from the

mobile stations and access points. In this case, the central server is used as a database that converts the RFID tag contents into useful information, such as the shopper's profile. Also it will contain information about each reader, such as the location of it and the applications that should be given to a student based on his profile in that section of the building.

Agents coordinate with other agents, forming a profile of a certain group of moving products. A profile of moving products can be created to identify and help the shopper the next time they shop. This way a customer's profile can be brought up on the next visit and promotions on products related to the type of products that person usually shops for can be offered.

One issue that is of concern in this system is called "jamming." This problem exists when many objects move at the same time and direction, making an RFID reader miss some objects. This issue is solved by the use of software agents.

The system has several drawbacks: (a) it scales poorly due to the limited range of passive RFID tagging, and (b) the RFID tagging involves significant installation and maintenance costs.

### 2.2.4 ActiveCampus

Active Campus [13] is a location-aware system that uses the concepts described previously about user location detection, that is, it uses the signal strength of beacons emitted from the access points of the wireless local area network. The system has an ActiveCampus Explorer that supports location-aware applications such as instant

messaging between students around the university's campus. The explorer also provides maps of the user's location with locations of nearby peers, activities, and landmarks. The system also offers an explorer called ActiveClass [14] that supports classroom activities; a student can ask the professor a question electronically over the PDA with full confidentiality. ActiveClass also provides the professor with other classroom activities such as polling and student feedback. The applications that are created serve basic HTML to ensure that any networked device can view the applications in a web browser. The server that is used is a standard web server with MySQL support applications.

ActiveCampus location sensing is limited to only WLAN access points, which means less accuracy over the entire campus. Information is stored on MySQL servers, meaning no presence in the use of context modeling methods which allows the system to be flexible and extensible enough to handle the wide range of context types, as well as the relationships between them.

### 2.2.5 SCAPE Architecture

When building a system that performs the requirement of a context or location level awareness, there is a need to take into consideration the middleware. The SCAPE Architecture [15] creates a pervasive campus information system, wherein a compact flash card reader is placed on the PDA and tags are placed on everyday use items in the campus. For example, an RFID enabled clock held close to an RFID enabled tablet PC will start the calendar application. Therefore, the role of the Sentient Communication Architecture for Pervasive Environments (SCAPE) middleware is to receive an object ID (i.e. tag ID) and user authentication. It will then authorize the user for the query from the

database to retrieve information and convert it into a suitable representation for the user. The user's PDA will provide the middleware with the ID of the object that it has scanned. The middleware will check the LDAP (Lightweight Directory Access Protocol) database for access rights. It will then query the ontology databases for relative information about the user's profile, campus site locations, information about applications, etc. The result of the query will be presented to the user in a suitable format. The disadvantage of the system is that it relies on passive RFID tags which are limited in reading range. Therefore a massive number of readers must be installed.

## 2.3 Location Prediction Models

Fast and accurate user location prediction techniques have become a major research effort recently. The importance of prediction techniques can be seen at both the network and service levels. Numerous attempts have been made to solve the problem of predicting user movement. Most of the attempts that address the issue of mobility prediction techniques are based on the use of historical movement patterns of the user but lack the use of other information, for instance environment and contextual information, such as user profile and preferences. The disadvantage in those models arises when the user is situated in new locations because the history of the user's movements is not available. Our model differs from those mentioned here by using the user's location history as well as environment and user contextual information, and combining this information with a rich mathematical theory of evidence as a tool to increase the accuracy in predicting the user's future location.

### 2.3.1 Location Prediction Through User's Movement Paths

Orwant [16] uses active badges, Unix logins, and schedule files, to increase the accuracy of location prediction by modeling the user's moving behavior through storing all the possible movment paths and related mobility patterns that are derived from the long term history of moving events of the mobile user. The limitation of this system is the large number of required example locations before the modeling system can be used. In Ashbrook et al. [17], the same concept stated above is applied, in addition to a GPS system that collects location information over time. Those locations are used as an input to a Markov model to predict the user's future location.

### 2.3.2 Location Prediction Simulation

Numerous simulation studies have been proposed. A simulation study was performed in [8] based on the use of contextual information. Uncertainty of the user's navigation behavior was captured by gathering pieces of evidence concerning different groups of candidate locations. These groups were then refined to predict the user's future location when evidence accumulates using the Dempster [18] rule of combination. The proposed approach did not impose any assumptions concerning the availability of a history of the user's movements. Another simulation study performed in [19] used data collected within ORL (Olivetti and Oracle Research Laboratory) to track the movement of its staff members for a period of 15 days using various prediction schemes. Roughly 60% of the data was used to establish a history database, and 40% of the data was used to carry out movement estimations. The results showed that the prediction accuracy of the system was between 50% to 70%. The limitations of this system is that it relies on user's mobility

patterns, so if there is a change in the user's behavior or if a user is situated in new locations there will be a failure in accuracy.

## 2.4 Context Modeling Methods

Location-aware systems require the use of context in order to achieve a high level of degree in detecting and predicting the location of a user and providing the required services. Therefore, a unified method of modeling and representation of context is required in this case. To make our system scalable and flexible to be used by other systems, it is required to have context information represented uniformly throughout the system. The system must also be flexible and expandable enough to handle the wide range of context types, as well as the relationships between them. Most systems today are distributed; it is, therefore, necessary for context models to be easily shared, especially in our architecture given its use of users' profiles. Models should also have a high level of formality and be able to represent existing context relationships. For these reasons, researchers have developed a number of different methods for modeling context information. Some of these models do not meet the requirements of our system and location-aware systems, while others can be used as the basis for our ontology. In the following subsections, we provide an overview of available context models.

### 2.4.1 Composite Capabality / Preference Profiles (CC/PP)

CC/PP [20] is a specification proposed by W3C as a profile representation language. CC/PP is a framework based on the Resource Description Framework (RDF), which is an XML based meta data description framework. CC/PP is focused mainly on describing

capabilities and preferences for wireless devices and mobile phones. The main goal behind CC/PP is for it to be used based on the available features within the computing environment, the user's requirements, preferences, and application capabilities, as well as the requirements of the computing applications.

The CC/PP specification defines a basic structure for profiles. CC/PP profiles are composed of two-level trees that list components and their attributes. Each profile has a number of components, and each component has a number of attributes. The types of components and attributes allowed within a profile are dictated by one or more schemata.

The main disadvantage of the CC/PP framework is the strict two level hierarchy. CC/PP does not capture and represent all context information such as accuracy and resolution present within pervasive environments.

An extension of the CC/PP framework was recently introduced by Indulska et al. [21] who extended the original CC/PP vocabulary to include a wider range of context information about Network Interfaces of devices, Quality of Services, Location, Disconnection Status, and Application Requirements. The extension also included several relationships and dependencies that grouped related components together in their own profiles. The main idea behind this extension was to decrease the limitation of the original framework.

### 2.4.2 Comprehensive Structured Context Profiles (CSCP)

CSCP [22] is based on the Resource Description Framework (RDF) wherein context information is expressed by using session profiles. CSCP has the advantage of

eliminating the need to define a fixed hierarchy, and also CSCP is able to interpret attributes according to their positions within the profile structure. A profile in CSCP describes all context information relevant to a client's mobile session, such as device profiles, user profiles, network profiles, etc. CSCP provides mechanisms to attach conditions and priorities to attributes to resolve potential conflicts between preferences and service capabilities. To update existing profiles, differential profiles are used that contain reference to the previous profiles and that override the attribute values.

CSCP faces many limitations when it comes to applying its concept to location-aware environments. The use of RDF syntax for profile representation makes it an unlikely candidate for context representation due to the existence of more powerful ontology languages such as OWL. CSCP is also limited by its vocabulary to a set of context concepts and relationships. Location-aware systems are very similar, if not more complex than context-aware systems. Therefore, the complex relationships existing within these systems make the process of modeling concepts and relationships unintuitive for system designers, which complicates the process of building a location/context-aware system.

### 2.4.3 IETF Media Feature Sets

IETF Media Feature Sets [23] was designed to allow for protocol-independent content negotiation. It specifies device capabilities and user preferences by unstructured attribute/value pairs. This informality by itself is a drawback and will not be used for our system. It specifies features of the supported and preferred content representation rather than the device capabilities. Complex capabilities and preferences are expressed by

Boolean expressions of attribute pairs. Media Feature Sets allows the possibility to assign quality values to capabilities and preferences descriptions. Another drawback to the framework is that it is not decomposable, and there are no formal, machine readable means for extensions.

### 2.4.4 Friend of a Friend (FOAF)

The Friend of a Friend (FOAF) [40] project is designed for creating a web of machine-readable pages describing people, the links between them and the things they create and do. The project focuses on the social and structural relationships involved in the ontology rather than the representation, development and use. FOAF provides an RDF/XML vocabulary to describe personal information. FOAF is a useful building block for creating information systems that support online communities, which enhances the way personal information and relationships are expressed. FOAF is useful in managing communities, expressing identity by allowing unique user IDs, providing assistance to new entrants in a community, locating people with common interests and much more. One drawback to this context modeling method is that FOAF makes partial use of OWL, making it difficult for many OWL tools to understand the FOAF ontology.

### 2.4.5 Ontologies

Since we chose Ontologies as our preferred choice due to the many advantages that we will discuss next, a more detailed description will be provided to have a thorough understanding of ontologies.

One of the most widely researched context modeling methods has been that of using ontologies. This is mainly because ontologies can represent concepts and relationships by employing a computer-usable data structure while also sharing a common understanding of the domains in which the modeled context can be easily used for Location Based Services (LBS). Ontologies are structured to be represented through a set of entities, functions, instances, and axioms. Ontologies are also known for their normalization abilities and formality, making them a favorable candidate for modeling context knowledge.

Most proposed ontology-based context models have adopted the OWL language as a means of ontology representation. This is due to OWL's superior expressive abilities over other languages, such as XML, RDF, RDFS, or DAML+OIL. OWL also allows for the exchange and comprehension of context information between different entities of a system. The availability of a number of OWL-based reasoning engines that can interpret context information present within the ontology or that can infer new context from existing context and relationships, makes OWL an especially effective language for context modeling.

In an attempt to propose a generic model on which all context models can be based on, Stimpakou proposed an ontology called COMANTO [24]. The proposal aims to enable various end-users, context providers, service providers, resource providers, and manufacturers to share a common vocabulary and to collaborate and synchronise their knowledge. COMANTO (Context Management Ontology) is a generic upper-level context ontology that is not specific to any domain, application, or condition. On the top level of the ontology is a class called Semantic Context Entity (SCE). This class

represents the root from which the most widely used concepts stem. Several subclasses that extend this major super class have been identified. These concepts have been categorized into 12 core or sub-classes as shown in Figure 2.1. The following is a brief description of these core classes.

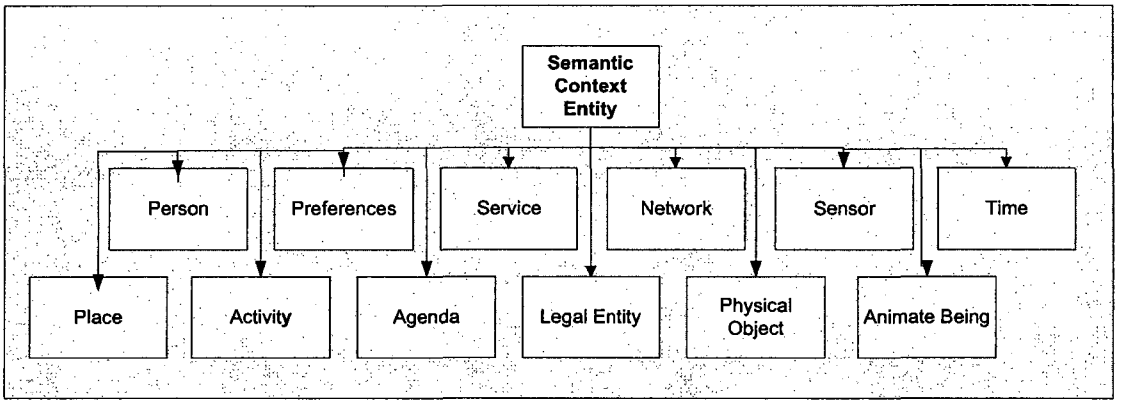



**Figure 2.1. The COMANTO Ontology Core Semantic Context Entities**

- Person class: This class corresponds to all human entities and offers various properties for incorporating the user related context, relationships, and behaviors.

- Place class: This class represents the abstraction of a physical spatial place. It is used to describe the physical location in terms of symbolic or geographic representation.

- Preferences class: This class represents user preferences in a generic manner. It is used to describe the user's device, service, and network preferences.

- Agenda class: This class describes the calendar information model of a user. This class is necessary to efficiently model the user's activity.

- Activity class: This class describes information relevant to the user's task during the specified task's duration. The class is divided into two types: the physical and the service activities. These two types model the user's tasks whether they are conveyed physically, such as working on a laptop, or part of a service, such as using a particular application.

- Time class: This class contains all necessary information related to the current time. It serves as a timestamp for all context information that may change over time.

- Physical Object class: This class is a super class of another class called "Device." The physical object class describes any object physically, except for devices.

- Sensor class: This class represents sensors that are used to collect context information. It contains dataype properties of sensors as well as configuration features.

- Service class: This class represents information to all services and applications to which the user subscribes.

- Network class: This class models context information related to the underlying network.

- Legal Entity: This class represents the corporate actors involved in the system.

The following, Figure 2.2, presents the associations and relations between the core classes of the system.

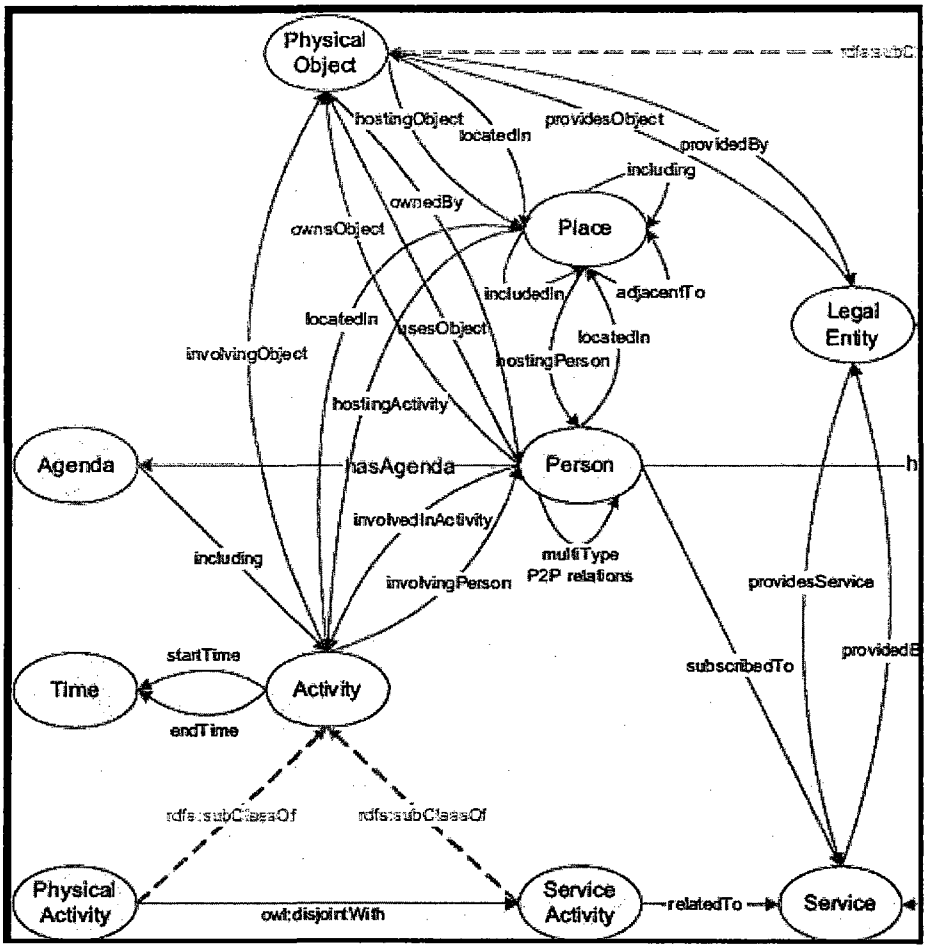



**Figure 2.2. Partial View of the COMANTO Upper-Level Ontology [24].**

Another attempt to propose a generic model on which all context models can be based on was provided by CONON [25]. OWL Encoded Context Ontology (CONON) was proposed to model contexts in pervasive computing environments and to support logic-based context reasoning.

CONON provides an upper context ontology that captures general concepts about basic context and also provides extensibility for adding domain-specific ontology in a hierarchical manner. CONON's domain-specific low-level ontologies can be dynamically plugged and unplugged from the upper ontology based on changes in the environment.

The upper ontology is broken down into four subcategories: person, location, computational entity, and activity. Details are shown in Figure 2.3.

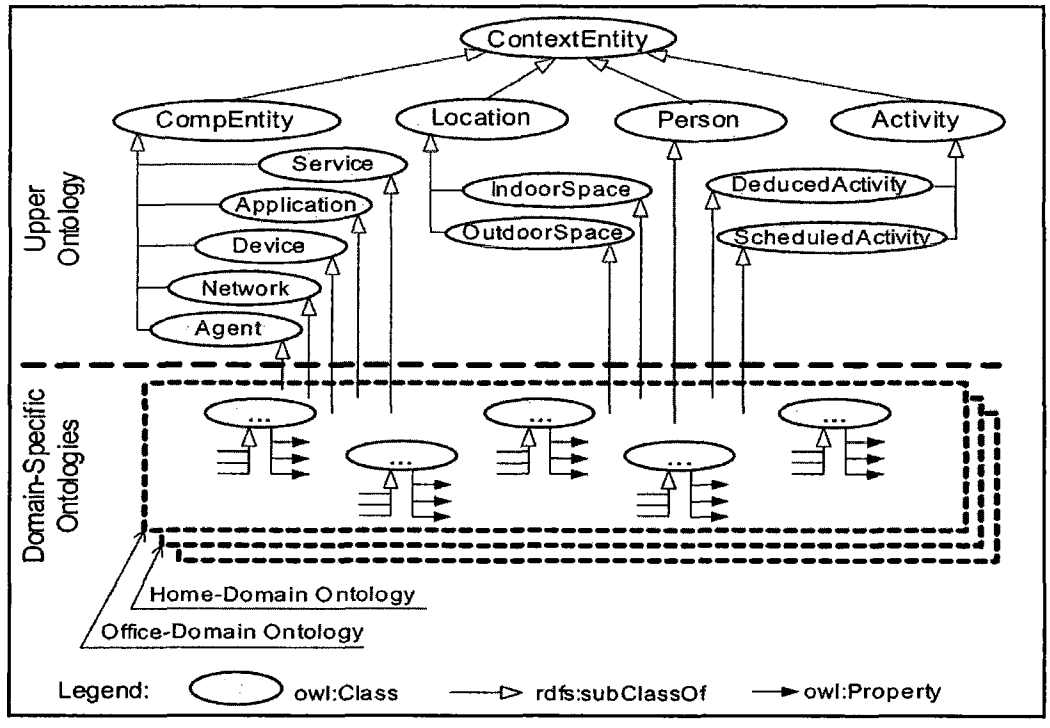



**Figure 2.3. Partial Definition of CONON upper ontology [25].**

Each entity is associated with its attributes and relations with other entities. The built-in OWL property '*owl:subClassOf*' allows for hierarchically structuring sub-class entities to provide extensions for adding new concepts required in specific domains.

The specific domains can be spaces like a home or office domain. Figure 2.4 provides a detailed explanation of a specific ontology for a smart home application domain.

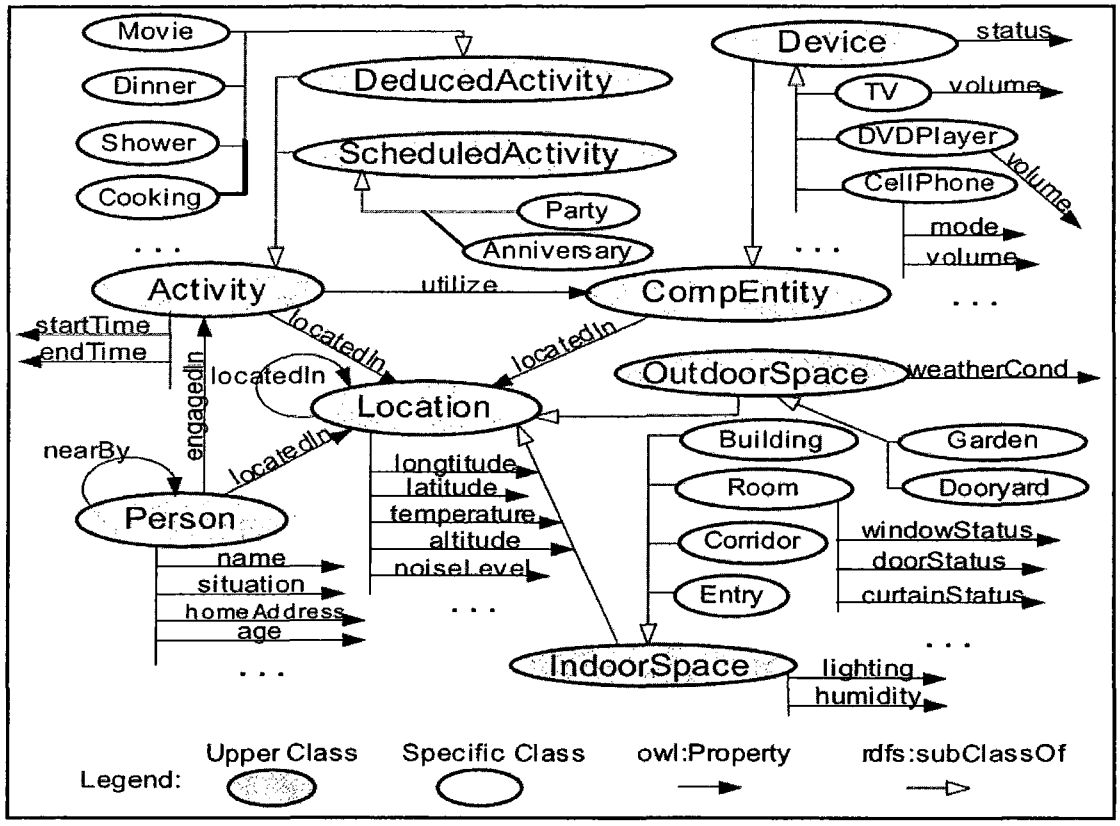



**Figure 2.4. Partial definition of a specific ontology for home domain [25].**

Another work titled "Service-Oriented Context-Aware Middleware" (SOCAM) [26] was developed by the same researchers. They have employed an OWL-based ontology. An interesting and useful contextual representation was added, which is a quality of context representation, allowing the ontology to extend the sensed context with quality constraints, such as accuracy and freshness. Each constrained quality is associated with a number of quality parameters, and each parameter is described by a set of appropriate quality metrics. The quality parameters are accuracy, resolution, certainty, and freshness.

## 2.5 Summary

In this chapter we have presented some of the available location tracking systems proposed by researchers in this field. Many of the proposed systems lacked the involvement of multiple technologies and the use of ontologies to combine location tracking with prediction. We viewed some context modeling methods suggested by researchers in the field of context awareness. Many of the proposed models lacked the ability to model complex context relationships, while others were interesting enough to be involved in our work and enhance it according to our domain. We also looked at the different location prediction systems involved in the research field. Our work will be based on the Dempster-Shafer theory expanding the work performed by Nancy [8] to include it in our location tracking method.

In light of the advantages and disadvantages of the models and system architectures discussed here, we will present in, Chapter 4 our method of building a system architecture for user location tracking involved with the use of an OWL-based ontology, discussed in Chapter 3, that provides an ontology that is richer in its content and is an extension of the work conducted by Yousif [6], adding the required context for the domain the system is involved in.

Details of the Dempster-Shafer theory and how it is used in our prediction method is discussed in Chapter 4. In Chapter 5, we present our detailed implementation of a prototype system illustrating the validity of our location-aware systems' architecture. We will also evaluate the system and compare the results with other systems.

# Chapter 3

## The Ontology: A Context Modeling Method

### 3.1 Objective

Our lives have become increasingly dependent on the availability and exchange of information. A Location-aware environment aims at presenting the right information to the right users, at the right time and in the right place. To be able to provide such services, a system must have a thorough understanding of its environment, the people and devices that exist within it, their interests and capabilities, and the tasks and activities that are being undertaken. For applications and services to operate efficiently within a pervasive environment, a common model must be created to represent the environment in which the users and devices are situated.

In chapter 2, we described the various approaches used to model context information. The modeling approach that will be used must have a high level of formality and an ability to represent the complex relationships between different context concepts. The model should be easily sharable and support the distributed composition of ubiquitous systems. We have chosen an OWL-based ontology conducted by Y. Al Ridhawi [6] to model all our context information within our location-aware architecture since it covers most of the required context. We have extended and integrated this ontology within our architecture. The remainder of this chapter is dedicated to describing the details of the original and extended ontology structure before introducing our system architecture in chapter 4.

## 3.2 Ontology

The Ontology represents the base onto which all context information within the location-aware system should be modeled. As seen in COMANTO [24] and CONON [25], ontologies were divided into an abstract high-level ontology and a lower domain-specific ontology. Using this concept of domain separation, our Ontology can be seen as our modified version of the Upper or High-Level Ontology presented in both of these works.

The modeled context presented in [24] and [25] provide high-level ontologies to create the most abstract set of classes and leave the remaining details for the domain-specific ontologies. In [24], the high-level context ontology was classified by the following set of classes: Person, Place, Preferences, Agenda, Activity, Time, Physical Object, Sensor, Service, Network, Legal Entity, and Animate Being. The details of the subclasses and properties descending from these high-level context classes were not provided for all. Only a small portion of inter-class relationships were given. In [25], the high-level context ontology was classified by the following set of classes: Person, Location, Computation Entity, and Activity. The details of the subclasses and their properties were left for domain-specific ontologies set according to the environment.

Designing domain-specific ontologies becomes more clumsy as the model gets into more details because many of the missing concepts and relationships must be created from scratch.

We created an upper ontology similar to what was mentioned previously, which we call *Ontology*. Within this *Ontology,* we attempted to contain the majority of contextual concepts that could possibly exist within any location-aware environment. Therefore,

many of the classes that existed within [6], [24], and [25] were imported and expanded; their hierarchical organization was reordered to allow for easier extensions within the low-level domain-specific ontologies, and an extensive list of inter-class relationships and properties was defined to ease the development of domain-specific ontologies.

Figure 3.1 shows the topmost classes within our Ontology. We will express each class's details in the following subsections, including object properties (i.e. relationships between classes) and data type properties.

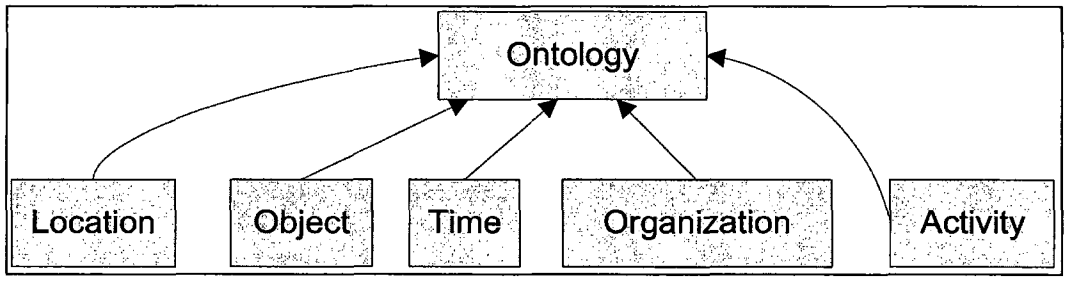



**Figure 3.1. The Upper-Level Ontology**

The *Ontology* class represents the root from which all context classes and properties stem. It provides an entry point for declaring all domain-specific ontologies. Five major classes extend from the *Ontology* class, each representing a major concept needed within location-aware systems:

- *Location*
- *Object*
- *Time*
- *Organization*
- *Activity*

Each class of the upper-level ontology is further divided into subclasses. For example, the Time class is divided into Relative Time, Exact Time, and Date. The following subsections provide details of the domain-specific classes.

### 3.2.1 Location Ontology

Location-awareness requires a data model that can adequately represent location. We refer to such a model as *Location Ontology*. The ever-increasing usage and computing power of mobile devices made knowledge of the user's location an essential element in adapting to the user's needs. By making such location-aware systems, we can increase application-space with the higher intelligence for different domains by providing context-dependent deployment and seamless services.

In modeling a location correctly, a great deal of context information can be induced, such as the activity taking place or the service to be provided in that specific location. Knowing that a person is at "home," in the "office," or in the "car" is often sufficient for applications to carry out predetermined actions in a given situation, such as turning off a cell phone's ringer during a meeting. In this case, the user's relationship or interaction with a place is more important than the physical location.

Most of the work that we have seen concerning ontologies does not provide any details on how "location ontology" should be designed, since this problem is mainly solved at the domain-specific level, it is left for the researchers at their specific domain to resolve. But, as we saw, location has been always at the most upper level of the ontology. Location is then divided into an indoor and an outdoor space. There are multiple ways of designing the location model. However, to make the process of defining location within

location-aware systems intuitive, we have divided the *Location Ontology* into two subclasses: *GeographicalPosition* and *SpatialProximity*. Figure 3.2 provides details of the subclasses branched from those two. Definitions and more explanation of each of the subclasses is given next.

#### *3.2.1.1 GeographicalPosition*

Locations represented by the address of a particular building, rooms of a building within which a user is located, or geographical positions such as the ones displayed on a GPS system are all considered GeographicalPosition. On the other hand, SpatialProximity refers to locations in relation to other nearby entities. GeographicalPosition is divided into five different classes: CoordinatePosition, OutdoorPosition, IndoorPosition, Distance, and Address Type.

The most obvious location representation, which is geographical coordinates, such as the one used in a GPS system, is represented within our ontology as "CoordinatePosition." It can be used to represent the global position of any entity along the Latitudes and Longitudes, as well as the entity's elevation in the Altitude subclasses.

When describing a place or space that is not under any boundaries of a house or building, such as parks, municipalities, cities, provinces, and countries, we refer to it as the "OutdoorPosition" class. This class is further divided into four subclasses: Continent, Country, Province, State, City, and Street. The use of these classes will help a system to perform certain services based on these external locations.

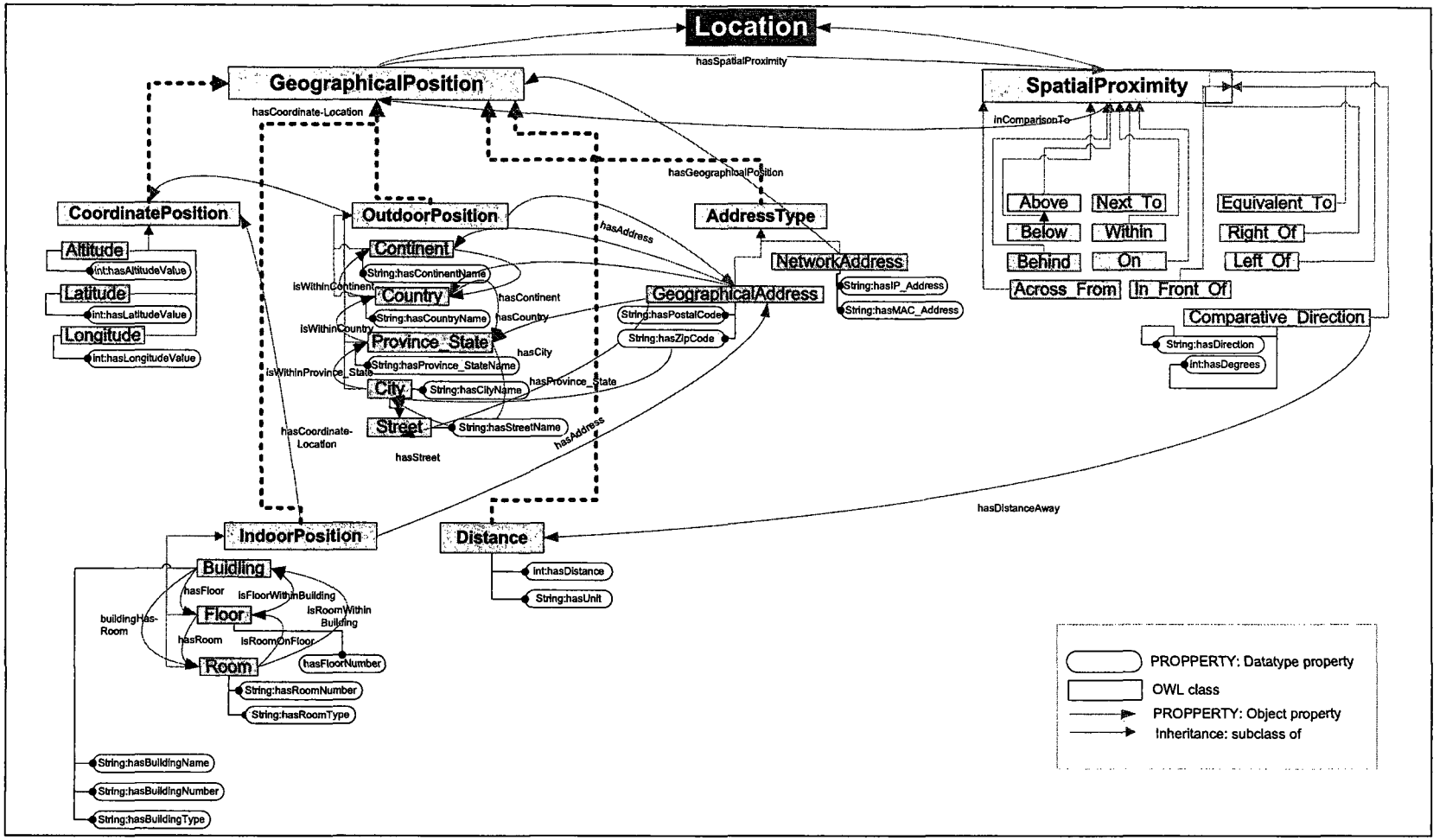



**Figure 3.2. Location domain-specific ontology.**

Many modern GPS systems provide information concerning positions of nearby buildings, stores, malls, and parks. Therefore, a solid location-aware system makes use of the entities mentioned above to provide services based on nearby locations.

The *OutdoorPosition* subclass is associated with the *AddressType* subclass. This class allows the location of a unit to be represented according to *Country*, *City*, *Street*, and *Postal* or *Zip Code*. These elements are grouped within the *GeographicalAddress* class, which is a subclass of *AddressType*. This extendibility dissmises any limitations embodied from using only the *OutdoorSpace* alone.

To emphasize the reusability of concepts that are present within our ontology, we provided several object-type properties that link the *GeographicalAddress* class to concepts already provided within the *OutdoorSpace* class in order to describe the geographical addresses of buildings. Thus, *hasCountry*, *hasProvince_State*, *hasCity* and *hasStreet* data properties refer to the country, province or state, city, and street information that the address contains. Additional information, such as the postal or zip codes, are contained within their respective data properties, *hasPostalCode* and *hasZipCode,* which are directly connected to the *GeographicalAddress* class.

We included representation of indoor locations in the ontology. The *IndoorPosition* class has been divided into three subclasses representing the three general levels into which indoor spaces are divided: *Building*, *Floor*, and *Room*. Object-type properties have been provided to model the relationships between these three subclasses. A floor can contain a number of rooms, while a room belongs to a floor. Similarly, a floor belongs to a building, while a building contains a number of floors. The *IndoorSpace* class has also

been given a relationship to the *GeographicalAddress* subclass under the *AddressType* class since any building's address would have information about the postal address it belongs to.

A network address serves as a unique identifier for a computing device on a network. These addresses are used for identity and communication purposes. One of the best known forms of network addressing is the Internet Protocol (IP) address. IP addresses consist of four bytes (32 bits) that uniquely identify all computing devices. Another popular form of address is the Media Access Control (MAC) address. MAC addresses are six bytes (48 bits) that manufacturers of network adapters burn into their products to uniquely identify them. We have adopted these concepts and grouped them in the *NetworkAddress* class. Since each device that has an IP or MAC address is also located within a corresponding physical location, such as a server room, we have provided a *hasGeographicalPosition* relationship that permits a network address to relate to other types of location whether they are *OutdoorSpace* or *IndoorSpace*.

We also included a distance subclass "*hasDistance,*" used to indicate the distance value and "*hasUnits,*" to indicate the distance of a location. This subclass will benefit the *ComparativeDirection* subclass which we will discuss in the next section.

These contextual entities, classes, and attributes modeled in our context, not only form the skeleton of context, but also act as indices into associated information. The objectives of our location context model includes modeling a set of entities and providing flexible extensibility to add specific concepts in different application domains left for designers of domain-specific areas.

#### *3.2.1.1 SpatialProximity*

We often refer to the location of objects in relation to other nearby objects (i.e. the *spatial proximity* of objects to each other). This way of describing location is more commonly used by humans. For example, when giving a location of a place we often use "next to" to describe a place, such as "I will meet you in the room 'next to' Ismaeel's office." We know Ismaeel's office is in Room B502. Therefore, the room *NextTo* it is room B503. This kind of high-level description can also be modeled in ontologies to give a better kind of description to location.

We have provided twelve possible choices for modeling SpatialProximity, all grouped under the *SpatialProximity* subclass: *Above, NextTo, Below, Within, Behind, On, Across_From, In_Front_Of, Equivalent_To, Right_Of, Left_Of,* and *Comparative_Direction.*

These subclasses are straightforward to understand, and no further details are required, but, as an example, the *Above* subclass can be used to describe floors inside buildings, the 5$^{th}$ floor is *Above* 4$^{th}$ floor. The *ComparativeDirection* class, on the other hand, has more ambiguity and needs clarification. We often refer to the relative location of an entity in terms of direction and distance, in number of degrees from another entity. For example, a coffee shop could be located 'North, 30° west, and 50 meters' from a mobile user. This relative location can facilitate human readability and clarity. The *ComparativeDirection* subclass addresses this concept. The *hasDirection* property is used to indicate the direction towards which the object is located, such as north, south, east, and west. The *hasDegrees* property is used to indicate the number of degrees the object is located away from the reference point, e.g. "south, *50°* east." Knowledge of direction is sometimes

insufficient without knowledge of distance. Therefore, the *hasDistanceAway* relationship has been added.

### 3.2.1 Time Ontology

Time in location-aware systems is crucial information. The time context comprises all information related to the current time and serves as a timestamp for all context information that may change over time. Time is used to notify users of important events, such as upcoming meetings and appointments that require the use of time. Not only that, but since our location tracking and predicting methods require the use of past location history of users to enhance the tracking and predicting algorithms, time will be crucial in this case to identify the time of these past locations. Therefore, entities within a pervasive computing environment should have a unified method for representing and understanding context information that is time-related.

Following the ISO 8601 [28] standard provides an advantage in computer and application usage and prevents the confusion or misunderstanding that can result from various existing notations.

The *Time* ontology, shown in Figure 3.3, is divided into three subclasses: *Date*, *ExactTime*, and *RelativeTime*. The Date subclass includes the following properties: *dayOfWeekNumber*, *dayOfMonthNumber*, *dayOfYearNumber*, *dayName*, *monthNumber*, *monthNumber*, *monthName*, *yearNumber*, and *weekNumber*. The international standard date notation is YYY-MM-DD and is represented in our ontology by the *yearNumber*, *monthNumber*, and *dayOfMonthNumber* properties of the Date class. Another notation of a date is the "ISO week-numbering year," which is slightly different than the calendar

year. For example, 2009-W07 means the 7th week of the year 2009. The *weekNumber* property under the *Date* subclass serves this purpose. This format is popular in manufacturing industries. The *dayName* and *monthName* properties provide system designers more flexibility in terms of modeling time through different formats.

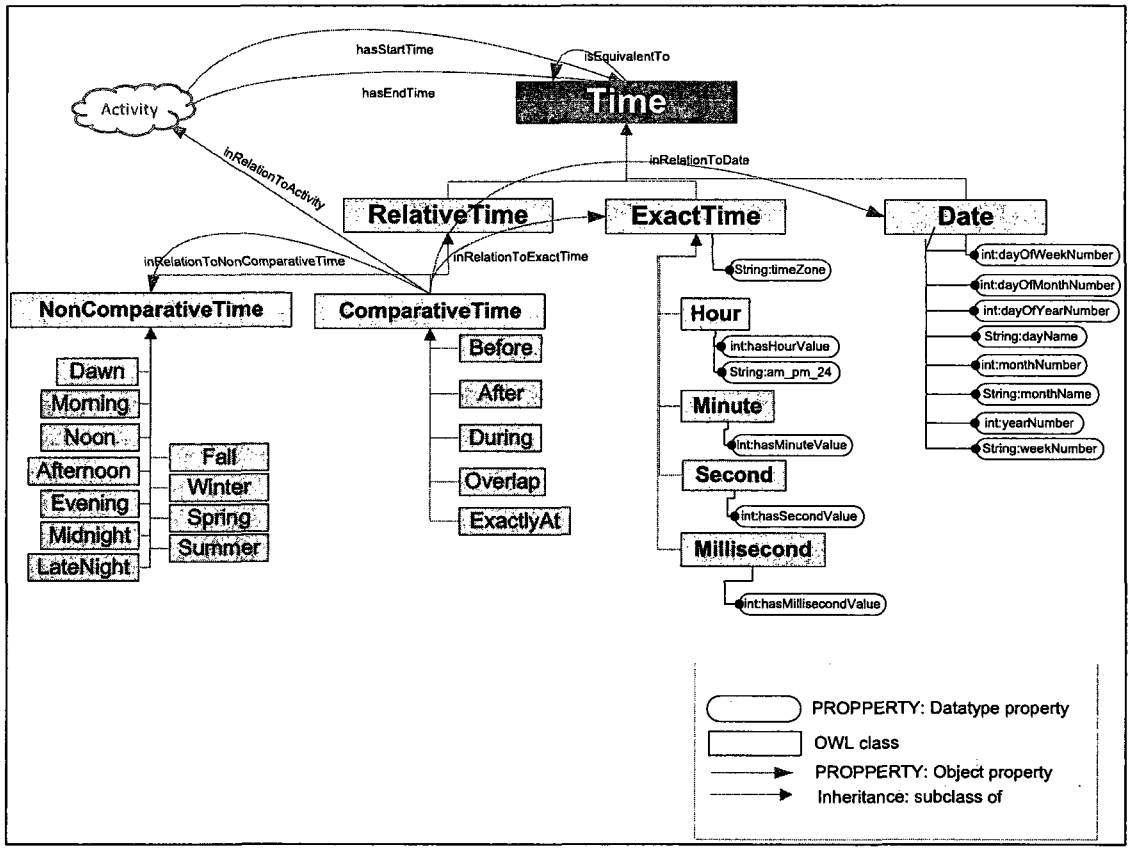



**Figure 3.3. Time domain-specific ontology.**

To model time according to the standard format hh:mm:ss, the *ExactTime* subclass is created. This class contains four other subclasses: *Hour, Minute, Second,* and *Millisecond.* Those subclasses include properties to exhibit their value representations. To

avoid confusion of the local time zone, such as GMT and EST, a *timeZone* property is included in the *ExactTime* subclass.

People tend to refer to time as a more abstract concept than that represented by the international standard. For example, the user 'Tom' could say that he has an appointment this 'afternoon.' The concept of 'afternoon' is understood to be between 12:00 p.m. and 6:00 p.m., but there is no reference to the exact time this appointment will take place. Such concepts of time have been grouped into a subclass of *Time* called *NonComparativeTime* as part of the *RelativeTime* subclass, which includes *Dawn*, *Morning*, *Noon*, *Afternoon*, *Evening*, *Midnight*, *LateNight*, *Fall*, *Winter*, *Spring*, and *Summer*.

People also tend to refer to time in another abstract concept, by comparing an event to time. For example, a meeting could be scheduled to take place 'after' an earlier meeting scheduled for 3:00 p.m. The duration of the first meeting is unknown. Thus the second meeting could have an infinite set of possible start times making the use of exact time representation inefficient. This is modeled under the *ComparativeTime* class allowing activities' exact times and dates to be compared to each other by using the following subclasses: *Before*, *After*, *During*, *Overlap*, and *ExactlyAt*.

### 3.2.1 Object Ontology

Since location-awareness is aimed at providing services to the user (i.e. the computing device carried by user), the presence of an ontology on which to model users and devices is, therefore, highly important. In addition, the use of RFID and WIFI devices to locate users proposes unique techniques in modeling the devices and signal strength received to

the end user. To take into consideration the wide variety of context available to be modeled around users and devices, we created a top-level domain class called "*Object.*" This class models anything that can be described as an object, such as a PDA, university student, RFID tag, WIFI access point, and Received Signal Strength (RSS). The *Object* class is divided into the following subclasses: *PhysicalObject*, *LogicalObject*, and *AtmosphericObject*. Details are shown in Figure 3.4.

The *PhysicalObject* subclass is the most complex one due to the wide variety of objects being classified as physical. Therefore, we divided this class into two more subclasses: *Person* and *Device.*

The process of modeling all possible users in all location-aware systems is a difficult task. For this reason, we have restricted our ontology to a set of limited OWL classes and properties that model the general concept of the *Person* class. The appropriate way of modeling users' information is according to the roles these users play within different domains. The needs and interests of users tend to change according to their role. As an example, a graduate student is permitted to access more information than an undergraduate student. A person in a company has a different role than a student or professor at the university. Therefore, *Person* class is split into three domains: *HomeDomainUsers*, *UniversityDomainUsers*, and *CompanyDomainUsers*. Although location-aware systems require other domains, we are interested in providing a subset of only those which are most widely used.

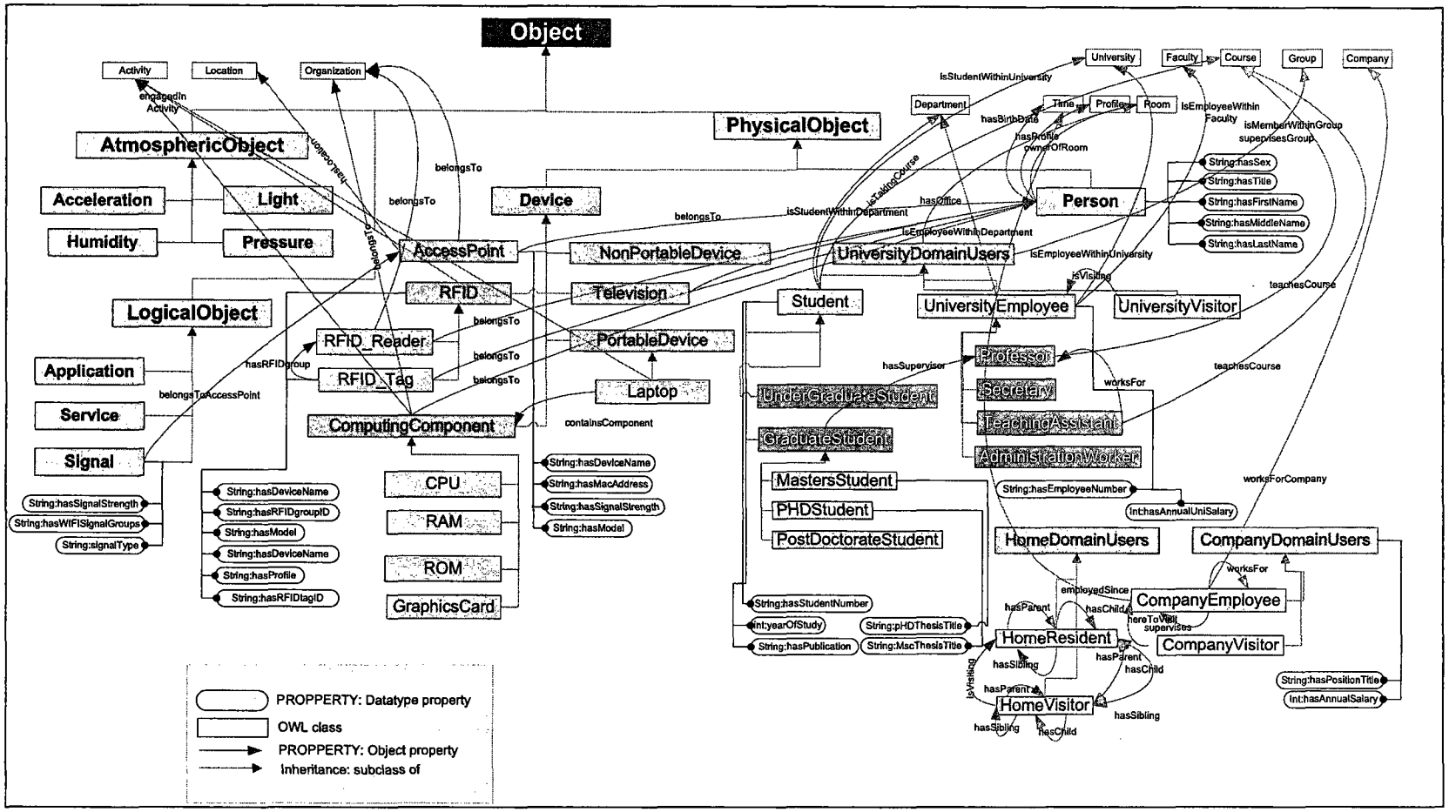



**Figure 3.4. Object domain-specific ontology.**

There always exists a set of context properties that all users share regardless of which domain they belong to. All Individuals could be divided into the categories male and female (*String:hasSex*); all human users have first and last names (*String:hasFirstName, String:hasLastName*), and all have a date of birth (*hasBirthDate*). Users could have many other possible common properties.

Most people have relationships with other individuals; we entitle these relationships in our ontology as *know*. Many classes have property relationships with different parts of the ontology, which is very helpful to us because it reduces the time spent creating ontology models from scratch. For instance, *hasBirthDate* links the *Person* class to the *Time* ontology, while *ownerOfRoom* links *Person* to the *Location* ontology, and so on.

The *HomeDomainUsers* subclass is used to model human users enclosed within the home domain. The needs, interests, and roles of home residents differ. Parents' context interests and accessibility rights are wider than their childrens', whose access permission may be limited. Also visitors are usually only permitted access to a limited amount of context information. Therefore, two subclasses are needed: *HomeResident* and *HomeVisitor*, and only four object properties are required: *hasParent*, *hasChild*, *hasSibling*, and *isVisiting*. All other context information can be inferred from these classes and properties.

Different classes of users exist in a university environment. In general, university users could be classified as *Student*, *UniversityEmployee*, or *UniversityVisitor*, with some individuals sharing two current roles, such as graduate students employed as *TeachingAssistant*.

Students are divided into two subclasses: *UnderGraduateStudent* and *GraduateStudent*. Graduate students are further broken down into three subclasses: *MasterStudent*, *PHDStudent*, and *PostDoctorateStudent*. This division is needed since access to many facilities within campuses differ for undergraduate and graduate students, and some of the information required for graduate students, such as the presence of a supervisor (*hasSupervisor*) or thesis topic (*String:MscThesisTopic, String:PhDThesisTopic*), is not necessary for undergraduate students. The same concept can be interoperated to *CompanyDomainUsers*.

It is important to break the Person ontology into different sections representing the domains in which clients can exist. Users in each domain should be categorized according to the roles they play in that domain, as we have provided in this ontology.

The process of modeling all possible devices in all location-aware systems is a difficult task. For this reason, we have restricted our ontology to a set of limited OWL classes and properties that model the general concept of the *Device* class according to our needs. We have classified the *Device* class according to five domains: a) *AccessPoint*, which refers to a router, switch, etc, b) *RFID*, which refers to the reader of that device, c) *PortableDevice*, referring to mobile computing devices such as laptops, d) *NonPortableDevice*, such devices are similar to desktop computers, f) *ComputingComponent*, which classifies objects such as CPUs, sound cards, video cards, etc, and finally, e) *Television*. Those devices belong to either organizations or persons; therefore, relationships are required in this case as seen from Figure 3.4.

Another important subclass of the *Object* class is the *LogicalObject* subclass. This subclass models context that is not classified as physical nor atmospheric. Applications and services provided by a system are considered logical. Therefore, any service that must be initiated once a location has been detected are all modeled and stored inside this class. Also, signals emitted from both WLAN and RFID access points are something that is not physical but rather logical. The reason we consider RSS values logical is because we have control over it with distance and obstacles between the access point and the receiving device. Since our location tracking method is based on the use of received signal strength (RSS) to calculate the location of a user, this class will provide simplicity when it comes to creating a location-aware user tracking and predicting system.

The *LogicalObject* class is divided into three subclasses: *Application*, *Service*, and *Signal*. The *Signal* subclass has a relationship with the *AccessPoint* class in order to identify to which router or RFID reader the emitted signal belongs.

Finally, the last domain of the *Object* class is the *AtmosphericObject* subclass. It refers to objects that are not physical. Such objects include: *Light*, *Humidity*, *Pressure*, *Acceleration*, and so on. Due to the wide variety of these atmospheric objects, we have limited our list to the ones mentioned above.

### 3.2.1 Organization Ontology

Different organizations have different hierarchical structures. The role played by the users of the system determines the organization they are in. A Master's student or a professor is part of a university, which is an educational organization, whereas an engineer at a high-tech firm is part of a business organization. Since this system is

intended mostly for users in these two organizations, we extended the *Organization* class into two subclasses: *EducationalOrganization* and *BusinessOrganization*. These subclasses can further be extended according to the domain-specific needs. Details are provided in Figure 3.5.

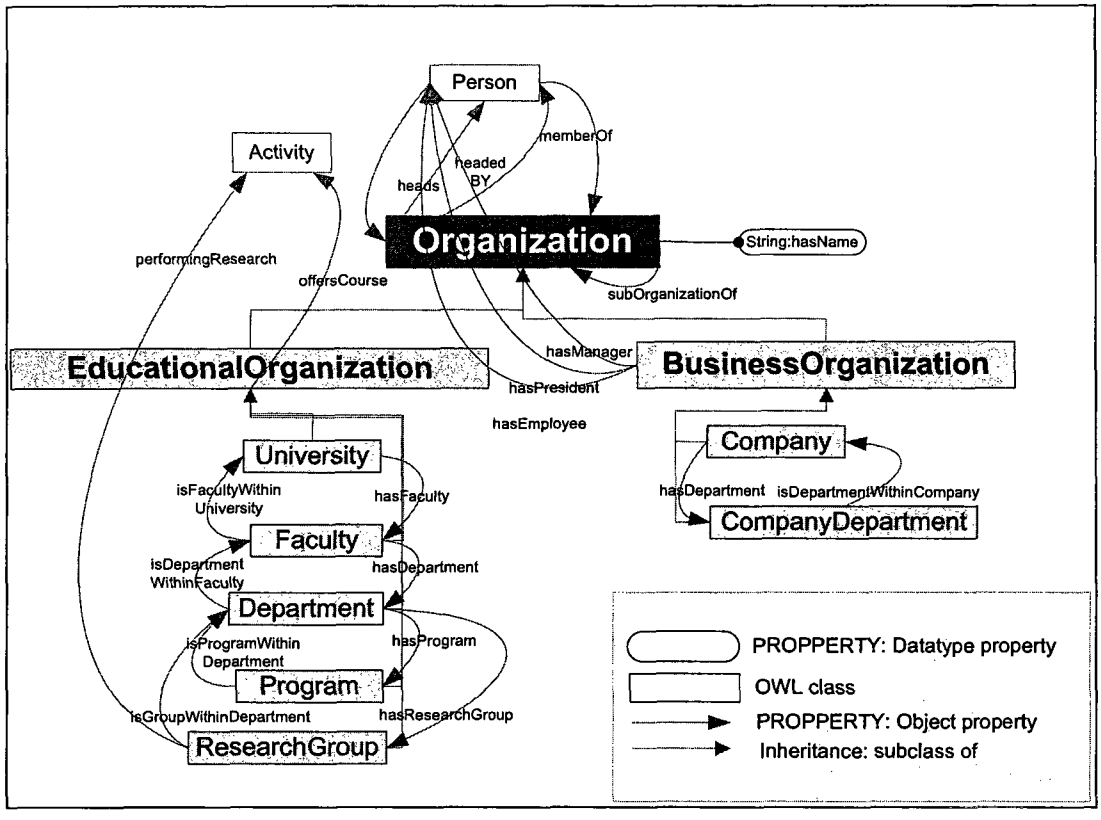



**Figure 3.5. Organization domain-specific ontology.**

We limited the educational organization domain to universities only. The *EducationalOrganization* class modeled the usual breakdown of universities into the following subclasses: *Faculty*, *Department*, *Programs*, and *ResearchGroup*.

The *BusinessOrganization* class represented the division of companies into various departments; therefore, the subclasses are: *Company* and *CompanyDepartment*. The

roles that exist within a company, such as manager, employer, employee, and assistant are represented as relationships: *hasManager*, *hasEmployee*, etc.

### 3.2.1 Activity Ontology

Many location prediction systems depend on activities conducted by users to identify the next location a user will occupy. Pieces of evidence can be generated from users' interests and activities to help predict the users' future location. Changes in the surrounding context may affect current or future activities, which may, in turn, affect the location prediction method. Therefore, our ontology includes a model for activities conducted by users. Figure 3.6 shows details of the subclasses included in the activity ontology and relationships that exist with other subclasses.

We have provided a list of possible activities such as *Walking, Meeting, Sleep*, etc. This list can be expanded to include other activities depending on the domain. This class has a relationship to the *Time*, *Person,* and *Location* classes. Most activities start and end at pre-set, inferred, or unknown times, and therefore require relationships to the *Time* class. Events may take place at different locations, which explains the need for the *locatedIn* and *hostingActivity* relationships with the *Location* class. Users participating or hosting events utilize the *involvedWithActivity* and *hostedBy* properties.

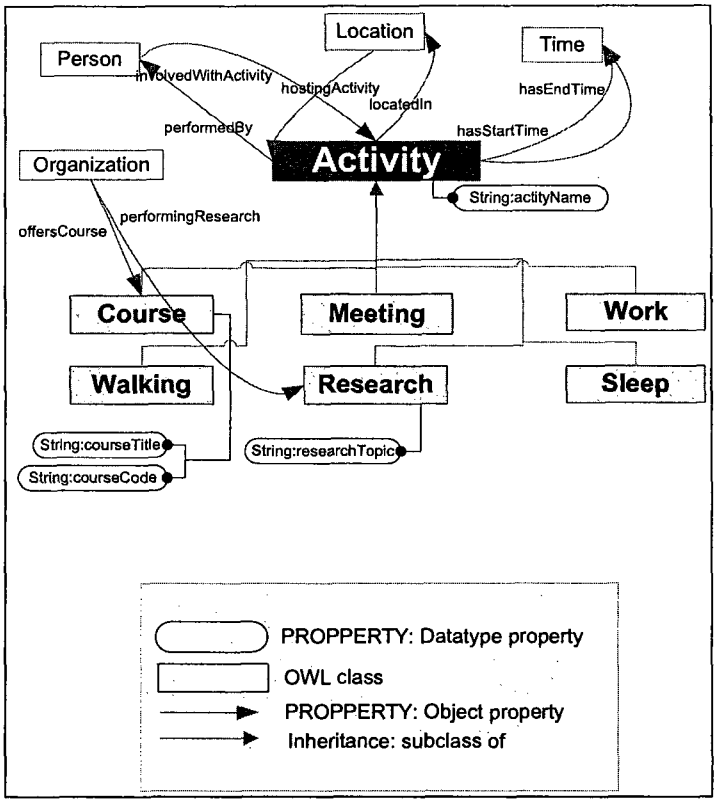



**Figure 3.6. Activity domain-specific ontology.**

## 3.3 Summary

Given the importance of ontologies for modeling knowledge and storing context information within location-aware systems, as well as the relative ease of sharing information between different entities within different systems, the ontology described above was designed to play a role in our system to better enhance capabilities in providing services required by different users based on location. Our ontology is not as abstract as those top-level ontologies presented by some researchers and is not too specific, i.e., only usable within a small set of predefined domains. Rather, our ontology

expands on upper-level ontologies provided in earlier models and generalizes the domain-specific ontologies seen in others.

This chapter described our proposed ontology with details of classes, sub-classes and relationships that exists inside the model. The following chapter will discuss details of the system architecture and how the ontology is incorporated and used within the location tracking and predicting system.

# Chapter 4

## Location-Aware System Architecture

### 4.1 Objective

The development of lightweight, portable computing devices and high-speed wireless local area networks have provided users connectivity while moving inside buildings. This inspiring paradigm has attracted many researchers to enhance applications and services that are a function of the mobile user's physical location. These kinds of technology advances in embedded systems are building the path towards the so-called ubiquitous society. Yet, these devices provide few services which are responsive to location; I want the office lights, temperatures, and electronics to be set according to the settings preferred moments before arriving. All my e-mails should be downloaded to my desktop once I arrive to the office. All incoming calls must be filtered according to office setting when I am present in the office. I want my list of books recommended by colleagues when I have time to browse and am at the library. All these requirements can be met if we have a Location-Aware User Tracking System.

Location-aware computing can be regarded as one of the most important elements of the ubiquitous intelligence. Due to the availability of location sensing, wireless communication, and mobile computing devices, location-awareness has been made possible. Distributed location sensors, such as Wireless Local Area Network , and RFID sensors, offer many new capabilities for contextually servicing the environment.

In chapter 2, we provided background information on existing system architectures and middleware in location-aware environments: RADAR, CampusSpace, MultiAgent, ActiveCampus, and SCAPE architecture in sections 2.2.1 through 2.2.5 inclusive.

RADAR lacked accuracy in areas of less than three access points. CampusSpace system provides interesting solutions, but it has several drawbacks. It scales poorly due to the limited range of passive RFID tagging, and it also has significant installation and maintenance costs. MultiAgent [12] is an RFID-based application that uses software agents to avoid jamming while checking out tagged objects through airports and supermarket gates. While this system provides interesting solutions, it does not focus on the problem of user tracking. The system has several drawbacks: (a) it scales poorly due to the limited range of passive RFID tagging, and (b) the RFID tagging involves significant installation and maintenance costs. ActiveCampus aims at solving the Location Based Services problem but lacks infrastructure in location sensors and context-aware modeling. ActiveCampus location sensing is limited to only WLAN access points, which means less accuracy over the entire campus. Information is stored on MySQL servers, meaning no presence in the use of context modeling methods which allows the system to be flexible and extensible enough to handle the wide range of context types, as well as the relationships between them. Finally, the SCAPE architecture uses ontologies to provide service to users based on their locations, but it lacks the use of good technologies to provide accurate and economical methods of location identification.

The remainder of this chapter is dedicated to describing the details of each part of the location-aware system architecture.

## 4.2 System Architecture

The goal of our system is to enable each user to interact with his or her surroundings based on his or her location. Systems of this kind can be used for ontology-based negotiations [6], seamless video handoff [7], and mobility prediction [8]. They can also provide basic services such as displaying a map of the surroundings to guide the user inside a building or printing a document on the nearest printer. The interaction varies as the location changes. For example, when the user is in his or her office, the mobile device may receive work data only, but, if the user moves to the cafeteria, data may include information on friends and entertainment. This requires the consideration of context that is described as characteristics that are physically and logically measurable, such as location, objects, services, applications, and the like.

Support for location-awareness must meet some requirements. A location model should be presented, and it should provide an expressive and flexible representation for the locations of mobile objects. Only essential knowledge of spatial properties of locations should be represented. The model must be suitable for graphical representation. An architecture must be created to support multiple sensor systems for location-awareness. The architecture must be scalable in terms of the number of applications and the number of mobile objects.

To understand how we can achieve our system, below we give an overview of how our system is architected. Figure 4.1 gives a view of the simplified location-aware system architecture.

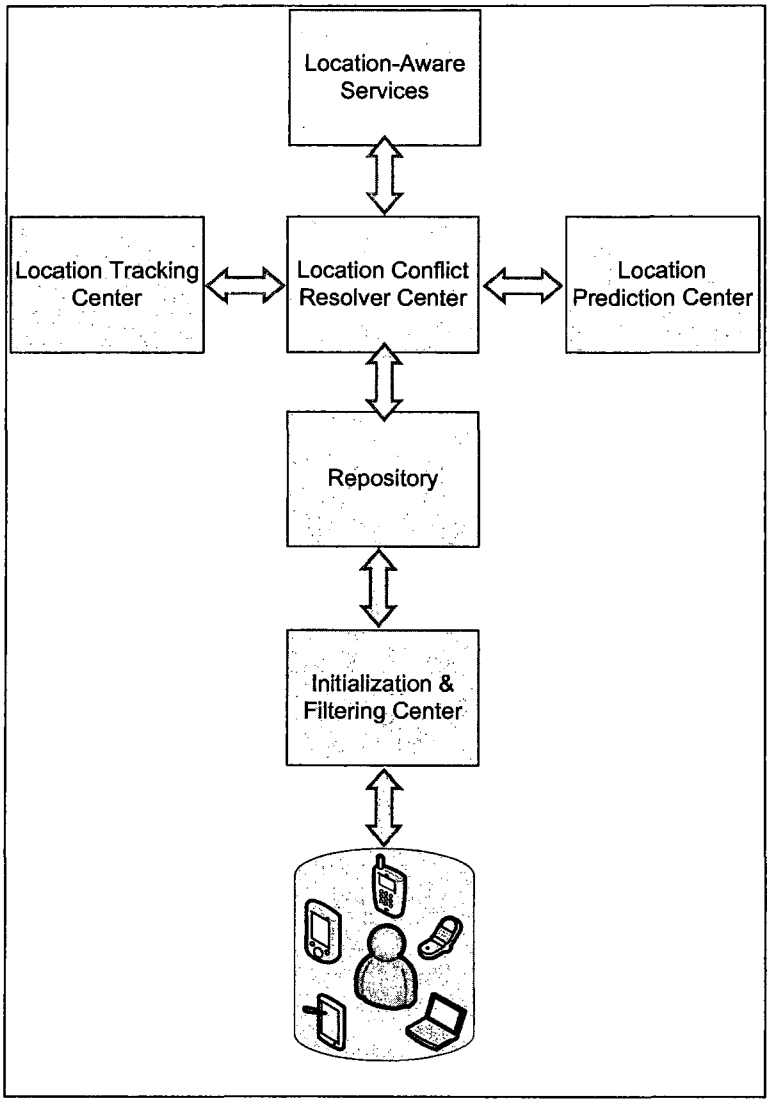



**Figure 4.1. Simplified System Architecture.**

Our system is divided into the following main parts: the "Initialization and Filtering Center," the "Location Tracking Center," the "Location Conflict Resolver Center," the "Location Prediction Center", and the "Repository."

The development of useful location-aware systems requires the establishment of universally accepted ontologies that permit the sharing of knowledge. We have created a

repository to store our data in an ontological fashion; it is integrated with the rest of the system.

The Location Tracking Center is responsible for obtaining the current location of a user, and it is identified by both WLAN and RFID technology. The signal strength emitted from both WLAN access points and RFID readers is obtained and processed separately. A full description of this Center will be discussed in Section 4.2.2.

The Location Conflict Resolver Center aims at resolving the conflict occurring between the two sets of location estimations, one from each of the WLAN and RFID tracking methods. If the two methods estimate conflicting locations, location conflict detection is triggered. Two conflict resolving solutions are proposed. The first chooses the best candidate from data stored in the location history repository. The second method is called "average location resolver." All the details of both methods will be provided in Section 4.2.3.

The Location Prediction Center relies on the Dempster-Shafer Theory. Once a user's current location has been identified, future location is predicted in order to provide better services. Our proposal is unique because it makes use of a user's "current-location identification." After an output from the Location Tracking Center is retrieved by the prediction center, context and history locations are both queried from the ontology. The Dempster-Shafer Theory uses these as inputs to its algorithm. A weight is given to each future candidate location, and the one with the highest weight is chosen. A detailed explanation of the algorithm will be provided in Section 4.2.5.

### 4.2.1 Repository

The Repository uses ontology-based context storage; its unified method of modeling and representation is flexible and extensible enough to handle both a wide range of context types and the relationships between them. We chose an OWL-based model for its flexibility and clarity. The repository is logically divided into the following classes: *user profile, context*, and *location history*. Those classes rely on the ontology that is further divided into more subclasses as seen from Chapter Three.

The *User profile* class classifies the following context in the ontology: information containing first, middle, and last name, date of birth, sex, student number, employee number, member of organization, daily, weekly, and monthly schedule, relation to other clients, regular activities, interests, people he knows, etc. The User Profile class relies on the Time, Object, Location, Organization, and Activity classes of the Ontology.

The *Context* class classifies the following in the ontology: information other than the user profiles, such as *GeographicalPosition* and *SpatialProximity* locations. A *GeographicalPosition* location is one specifically identified by a room number, floor number, building, address, and so on, while a *SpatialProximity* location is identified by subclasses such as NextTo, Above, and Below. Room B502, for example, is NextTo room B503. The repository also stores physical and logical objects. The physical object class includes information on all access points, RFID tags and readers. The logical object class contains the map of the recieved WLAN and RFID signal strength from access points and routers throughout the building. Other information, such as services provided in the current location and activities are also stored here. The Context class relies on all subclasses of the ontology.

The *Location History* class keeps a record of previous calculated locations. There are two reasons for keeping this type of context. First, for "location conflict resolving," that is, to significantly reduce calculation error by choosing the best of several possible locations, and, second, for location prediction. The Location History class relies on the Object, Time and Location classes of the ontology. Figure 4.2 illustrates how the Repository/Ontology is integrated with the overall system structure.

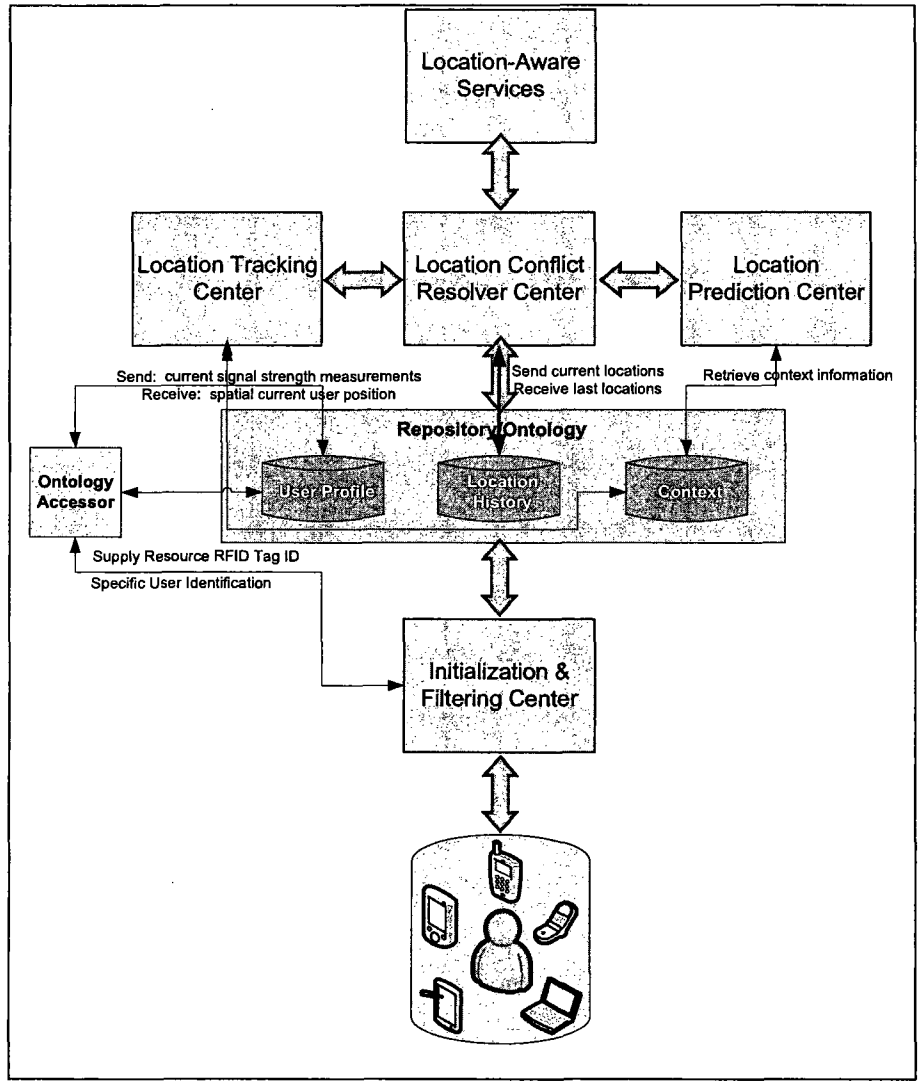



**Figure 4.2. Integration of the Repository/Ontology with the system.**

### 4.2.2 Location Tracking Center

Location-awareness in mobile environments is an important research area with a number of approaches. Work in the area makes use of different technologies to locate and track an object, including infrared, radio frequency, ultrasound, magnetic fields, and cellular systems.

Rao et al. [29] describe platforms, technologies, and standards used for location-based services (LBS). Technologies such as PDAs need to be connected and integrated with other infrastructures such as wireless and satellite networks to achieve success. Knowledge of the user's location is only one part of the problem. LBS also requires the time of day and information on how and why customers are navigating in a particular place. LBS is widely criticized on grounds of customer privacy, but the high demand for LBS systems shows that location-based services are a necessity in today's high-tech world.

We explained earlier that RADAR is a system that uses radio frequency to determine user location. This system locates users connected to in-building radio frequency WLAN by using signal strength information from the wireless network interface. The system provides accurate location information only when at least three access points are deployed close enough together that their coverage areas significantly overlap.

Active Bat [30] is a location system based on ultrasound technology. In this system, users carry tags that emit an ultrasonic pulse to receivers mounted on the ceiling. Each ceiling sensor measures the time for the ultrasonic pulse to arrive in order to compute its distance from the bat. The system can then calculate the location of the user wearing the tag. The

system has several limitations in scalability and in the installation and maintenance costs required for the large sensor infrastructure in the ceiling.

We also mentioned a system based on radio frequency reported in [12]. This is an RFID-based application that uses software agents to avoid jamming with moving tagged objects in airports and supermarket gates. While this system provides interesting solutions, it has several drawbacks: (a) it scales poorly due to the limited range of passive RFID tagging, and (b) the RFID tagging involves significant installation and maintenance costs.

ComMotion [31] is a system based on GPS. It gradually learns about the locations in its user's life from the user's daily travel patterns and then links personal information to those locations, thereby creating a location-aware environment. Other systems, such as the ones reported in [32] and [33], are also based on GPS. But, while this technology is useful outdoors, GPS receivers cannot detect satellite transmissions indoors.

Our work differs from those mentioned above. We attempt to solve the problem of location tracking with WLAN and RFID technologies with the integration of ontology. The following sections will discuss the details of the procedures.

#### *4.2.2.1 WLAN Location Tracking*

The fundamental idea in our WLAN Location Tracking algorithm is that the Received Signal Strength (RSS) is a function of the mobile user's location. It provides a means for inferring the user's location. There is a difference in the RSS values around inside a building. The signal received at the mobile device is strongest when the device is close to the access point and weakest when it is far away. It is advantageous to place a higher

degree of confidence in stronger signals and weigh them accordingly. A signal strength obtained closer to an access point is more accurate because the change in signal strength is greater as the device moves away. This observation can be applied to all available access points to estimate the mobile user's location.

Taking this method into consideration, our system takes the following approach to location determination. A WLAN signal strength map of the building is created, which stores locations in the building and the signal strength of the beacons emanating from the access points as observed at those locations. This WLAN signal strength is stored in the Ontology/Repository as shown in Figure 4.3 for clarity.

```
<Signal rdf:ID="B502_signal">
    <hasWIFISignalGroups rdf:datatype="&xsd;int">2</hasWIFISignalGroups>
    <signalType rdf:datatype="&xsd;string">WIFI</signalType>
    <belongsToRoom rdf:resource="#CBY_B502"/>
    <belongsToAccessPoint rdf:resource="#B502_MPRL802.11G-81"/>
    <belongsToAccessPoint rdf:resource="#Window_liksys-80"/>
  </Signal>
<AccessPoint rdf:ID="B502_MPRL802.11G-81">
    <hasDeviceName rdf:datatype="&xsd;string">MPRL802.11G</hasDeviceName>
    <hasMacAddress rdf:datatype="&xsd;string"
      >00:13:46:fa:79:20</hasMacAddress>
    <hasModel rdf:datatype="&xsd;string">0523</hasModel>
    <hasSignalStrength rdf:datatype="&xsd;int">81</hasSignalStrength>
    <hasLocation rdf:resource="#CBY_B502"/>
  </AccessPoint>
```

**Figure 4.3. Radio Map stored in the Ontology.**

From the figure, we can see that each location, in this case room B502 in the CBY building, has two available access points from which a mobile device can receive a signal. Therefore, our method of localizing a mobile user using IEEE 802.11 technology is divided into two phases. The first is the offline or training phase, wherein a WLAN Signal Strength Map of the environment is built by sampling the space and gathering data

at various predefined checkpoints of the indoor environments. In our case these checkpoints are modeled to be the door fronts in the hallway. Later, the mobile user walks in the same workspace, and the system locates and tracks the user's position.

Much of the effort in estimating the location of a user goes into creating the WLAN signal strength map of the building. To do this, a mobile user walks to several different locations in the building and records each location together with the signal strength of the beacon packets from each of the Access Points within range.

Our WLAN Location Tracking Center uses NetStumbler [34] to gather information about the available access points and the signal strength emitted from each one. The values are compared with those stored in the context Ontology to provide the equivalent of a WLAN signal strength map. As mentioned above, this map is created as a mobile user walks through the relevant area in the building, recording location coordinates and the signal strength emanating from each of the available access points. Figure 4.4 visualizes the WLAN Signal Strength map.

Studies [10] have shown that having more access points with overlapping coverage (see Figure 4.5) makes the estimation of a location more accurate. This is because system performance is improved, and there is added protection against downtime in the event of access point failure.

Once the offline process of information gathering is complete, live or real-time user tracking is initiated. Signal strengths from the available access points are retrieved and compared to those stored in the context Ontology. Once a location has been estimated, it is stored in the location history repository.

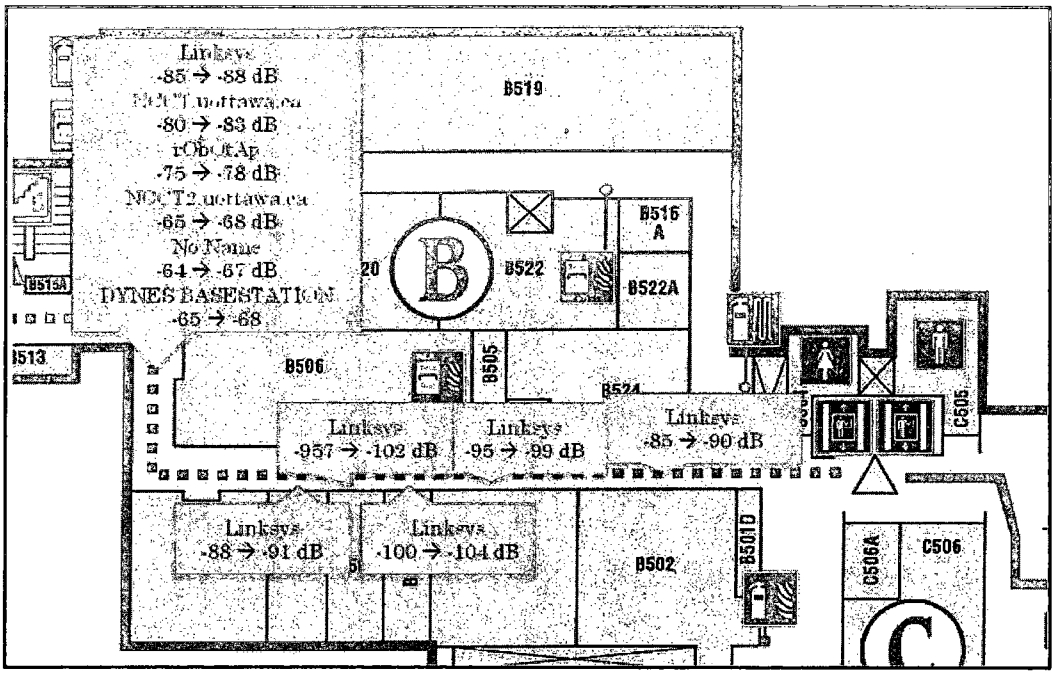



**Figure 4.4. Partial WLAN Signal Strength Map of the CBY 5th floor.**

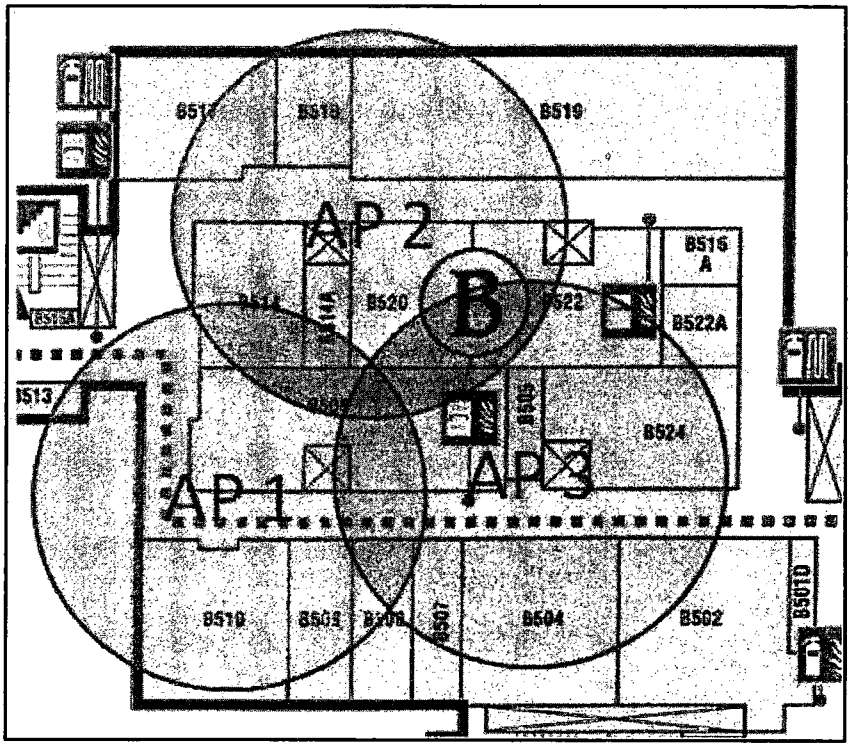



**Figure 4.5. Overlapping Coverage of Access Points.**

This concept is the same as triangulation [35], which is based on signal strength used to determine the location of a user. For every location, there is a unique reading of signal strengths gathered from a group of access points. There is always the need for an access point to exist; therefore, a minimum of one access point is necessary to determine location, with the addition of extra access points to increase accuracy.

A problem that exists when using wireless location area networks to detect the location of mobile devices is that the transmitted signal generally reaches the receiver through a multipath, meaning that the signal does not travel from the access point to the receiving device in a line of sight; rather, it travels through the area by reflecting from one object to another until it reaches the destination device. Sometimes the number of people in the building may distort the signal's travel and produce a distorted version of the transmitted signal. Due to reflection, refraction, diffraction, and the absorption of radio waves by structures and people inside a building, the WLAN Signal Strength map will not be accurate due to the fact that a map created at a particular time may not accurately reflect the environment at a different time. Figure 4.6 visualizes the problem.

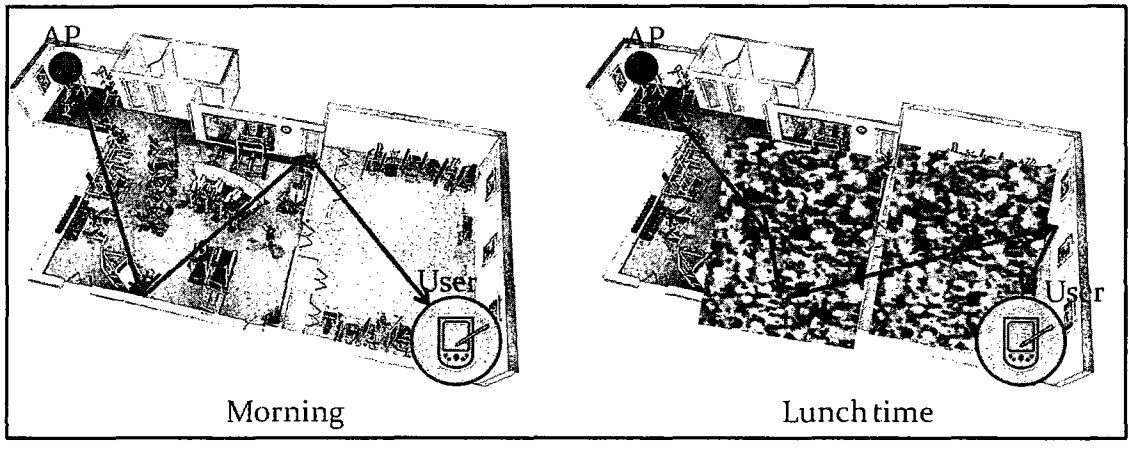



**Figure 4.6. Changes in the environment of a WLAN Signal Strength map.**

The solution to this problem is to use multiple WLAN Signal Strength maps to reflect different environmental states. Since each access point is fixed to one location, each access point will listen from other access points that are within range and record the corresponding signal strength. Any major change in the signal strength is due to a shift in the environment. At this time the access points will switch to the corresponding radio map. We have limited our system to one map since we are not limited to the use of WLAN alone, rather RFID will be involved to increase the accuracy of the system. There are other issues involved with WIFI signals, such as "signal aliasing." These issues and their solutions will be looked at in the Location Conflict Resolver Center section.

IEEE 802.11 standard (WLAN/WIFI) is attractive for location tracking systems since it is license-free. The fact that most off-the-shelf mobile devices are equipped with wireless Ethernet has also attracted the use of this technology in location detection. The next subsection will discuss the use of RFID technology in our system for mobile user location tracking.

#### *4.2.2.2 RFID Location Tracking*

The second algorithm used for user location detection is with the help of RFID technology. Many of the location-sensing systems developed today are based on radio. Early use of RFID was limited in competing with the barcode system that is printed on paper. The RFID technology was expensive in its early use, and was one of the limitations to its use. But today, with the price of RFID's being very low due to the shift in technology, researchers are able to develop very important uses for RFID. Some of the

uses of RFID in the research field are its involvement in Location Awareness and mobile user tracking.

A simple RFID architecture is constructed of three components: a) The *Tag* that identifies a specific object such as a PDA, Laptop, or person. RFID tags are classified into two classes: *Active* RFID devices and *Passive* RFID devices. The difference between the two is that active tags require a power source and have long range reading distance, while passive tags require no maintenance and have an indefinite operational life, since there is no need of a power source, but they have a short range reading distance. b) The *Reader* that reads whatever data edited or preinstalled on the tag, and c) The *Back-end System* which is a database that converts the tag contents into useful application entries.

Our RFID Location Tracking algorithm uses RF Code [36] RFID readers and active tags. The product operates on a frequency of 433.92MHz, giving a range of up to 150 feet. RF Code active tags follow RF Code packet format as shown in Figure 4.7. The format of the messages returned by the reader includes:

- Tag ID
- Group ID
- Timestamp
- Signal Strength
- Motion/stationary

Signal strength values are used to help identify the location of a tag. Each tag has a unique ID that distinguishes it from other tags. All RF Code tags are defined as being members of a specific group (i.e. Group ID) and have a unique tag ID number within that group (i.e. Tag ID).

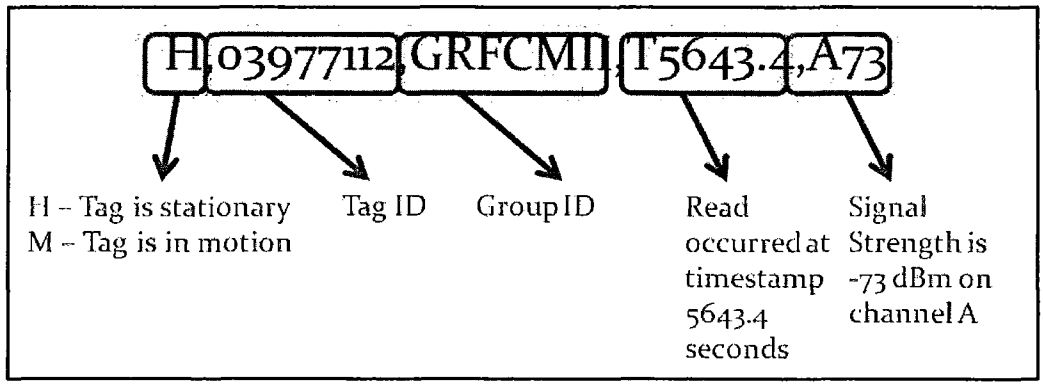



**Figure 4.7. RF Code Packet Format.**

The readers are placed so that coverage of beacon packets overlaps, as previously shown in Figure 4.5. Each reader is responsible for its zone, where it signs the user's RFID tag in and out as the client moves into and out of its coverage area. When a client enters a coverage area, the profile is obtained from the repository and tracking begins. Figure 4. 8 shows a section of a building with two RFID reader zones, each responsible for signing users in and out of its designated area. The figure also shows how servers are spread through the campus network. A more detailed picture is visualized in Figure 4. 9.

One very important topic that will be involved in our future work to help in location prediction and future service enhancement for mobile users is Software Agents. Agents can coordinate with other agents, forming a profile of a certain group of moving students. This will identify each student and his or her interests. For example, when a class is to begin, students will sign out from one location in the building and sign into another location. The students will form a profile with the time so that maybe next time the system can predict the location where the student is to be available and provide them with the needed context before arriving to the class (concept can also be visualized through Figure 4. 9 too).

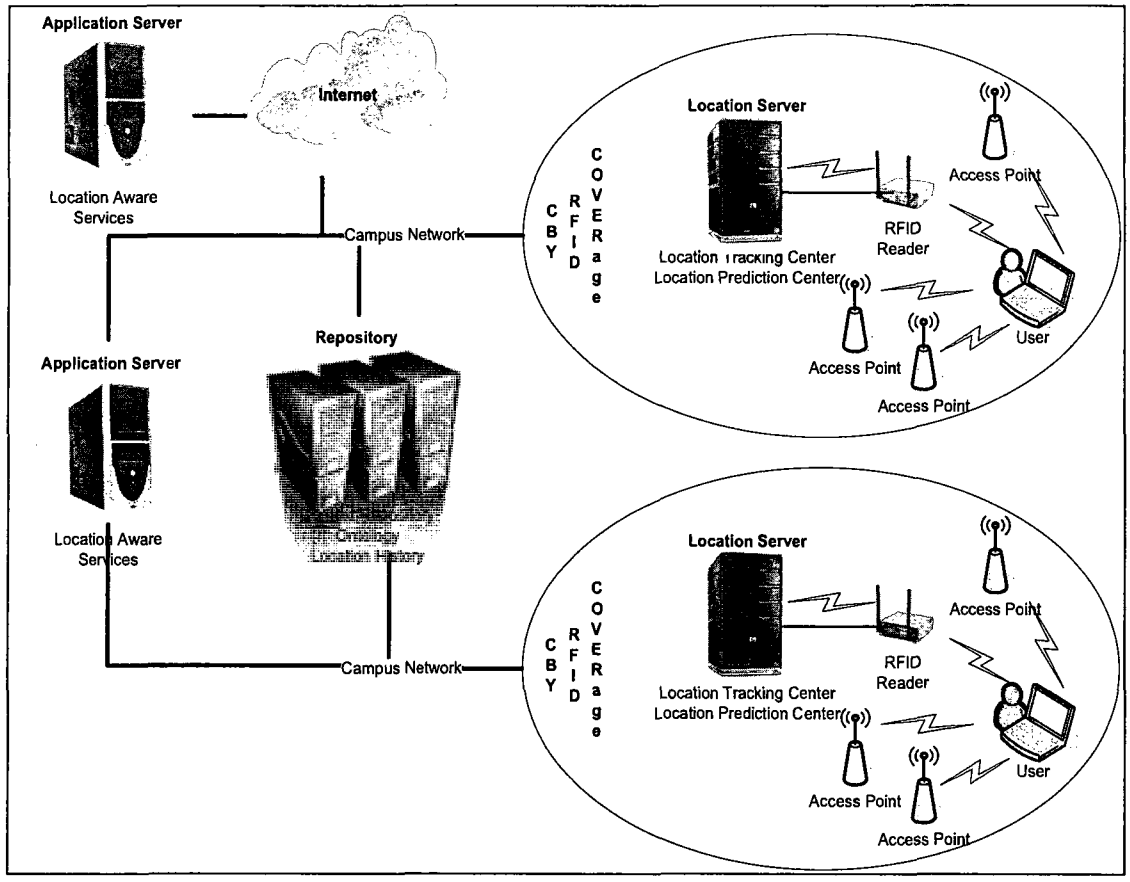



**Figure 4. 8. RFID Readers' coverage areas.**

The same concept can be applied to other areas, such as shopping at a grocery store. The software agents will be initiated once products are checked-out from one zone to another. Also a profile of moving products can be created to identify and help the shopper the next time they shop. In this way a customer's profile can be brought up on the next visit and promotions of products related to the list of products for which a person that usually shops can be offered.

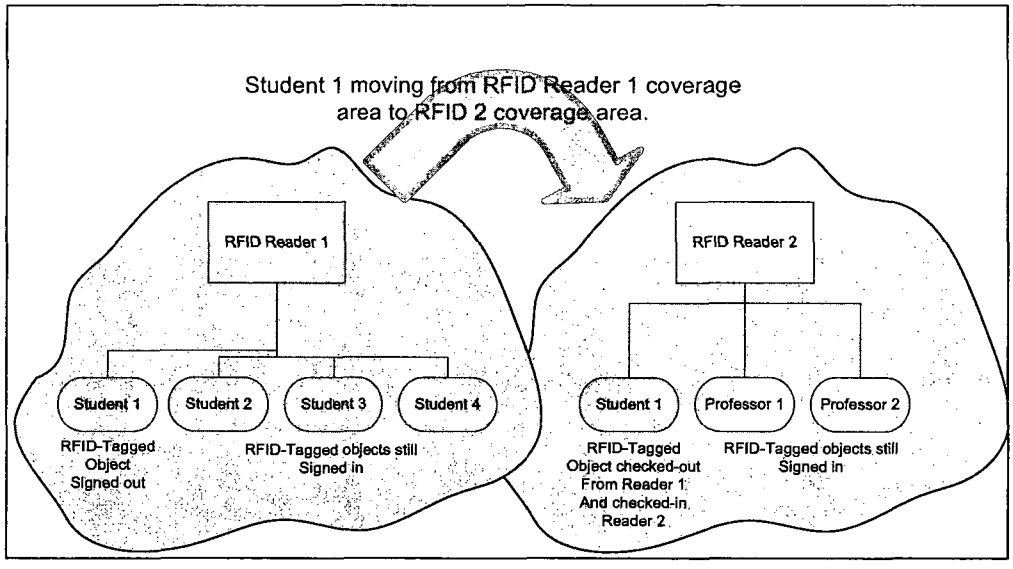



**Figure 4. 9. Detailed illustration of RFID Readers' coverage areas.**

The algorithm in which a mobile user's location is identified is very similar to the WLAN location tracking method used: the signal strengths emitted from the RFID reader are gathered and plotted onto a RFID signal strength map. Those values are stored in the context repository where they are compared to provide the user's estimated location. The locations are also stored in the location history repository for future use.

#### *4.2.2.3 Location-Tracking Center Overview*

In section 4.2.2, we have provided a detailed description of our Location Tracking Algorithm. We believe that the development of useful location-aware systems requires the establishment of integrated location sensing technologies. We introduced a location tracking system that makes use of both WLAN and RFID technologies to identify a user's current location. Our system architecture better serves the client by using location

and context information using a universally accepted ontology that permits the sharing of knowledge.

As an overview of how the Location Tracking Center operates, a simple example is demonstrated. Shown in Figure 4. 10, a student entering a building on campus is checked into the coverage area of an RFID reader that is placed near the entrance of the campus. The student's identification number identified using the active RFID tag is sent to the system, and the user's profile is retrieved using the Ontology Accessor. The wireless interface card gathers information on signal strength of beacon packets emitted from different access points using Netstumbler. Those values are sent to the location tracking center for new location calculation. The values are compared with the WLAN signal strength map stored in the context Ontology. A location is estimated and stored in the location history repository. The process of location estimation using the RFID data estimates a location in the same manner discussed for WLAN location estimation, wherein RF signal strength values are retrieved to be compared with the RFID signal strength map stored in the context ontology. A location is estimated and stored in the location history repository. Conflicts are dealt with in the following section. The process of location identification is repeated dynamically to determine the user's current location.

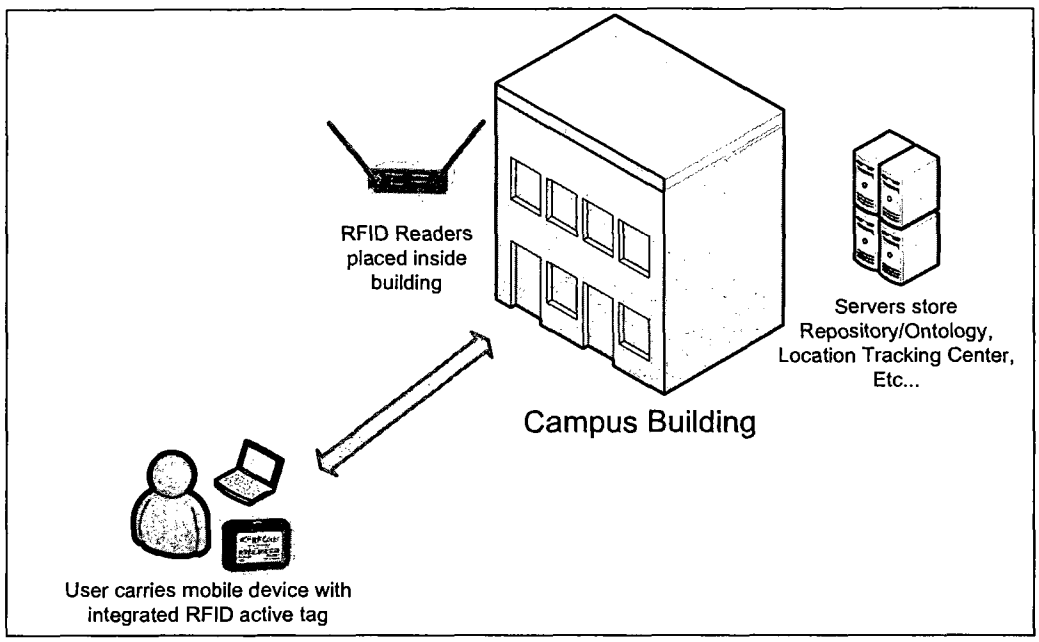



**Figure 4. 10. Simple example of using the Location Tracking Center.**

Figure 4. 11 illustrates how the Location Tracking Center is integrated with the overall system structure. Both WLAN and RFID tracking components are composed of three classes: *Signal Strength Retrieval, Signal Strength Interpreter*, and *Location History Value Updater*. The *Signal Strength Retrieval* class communicates with the Initialization and Filtering Center to retrieve the signal strength of beacons emitted from the Access Points and RFID Readers and received at the users' mobile device. *Signal Strength Interpreter* class communicates with the repository/ontology to calculate the current location of the mobile user. The *Location History Value Updater* class is responsible for updating the ontology to store the most recent location visited by the mobile user. If a conflict exists between RFID and WLAN in location estimation, the problem will be solved by the Location Conflict Resolver Center. If no conflict is present, the value is sent to the Location Prediction Center to predict the next locations of the user.

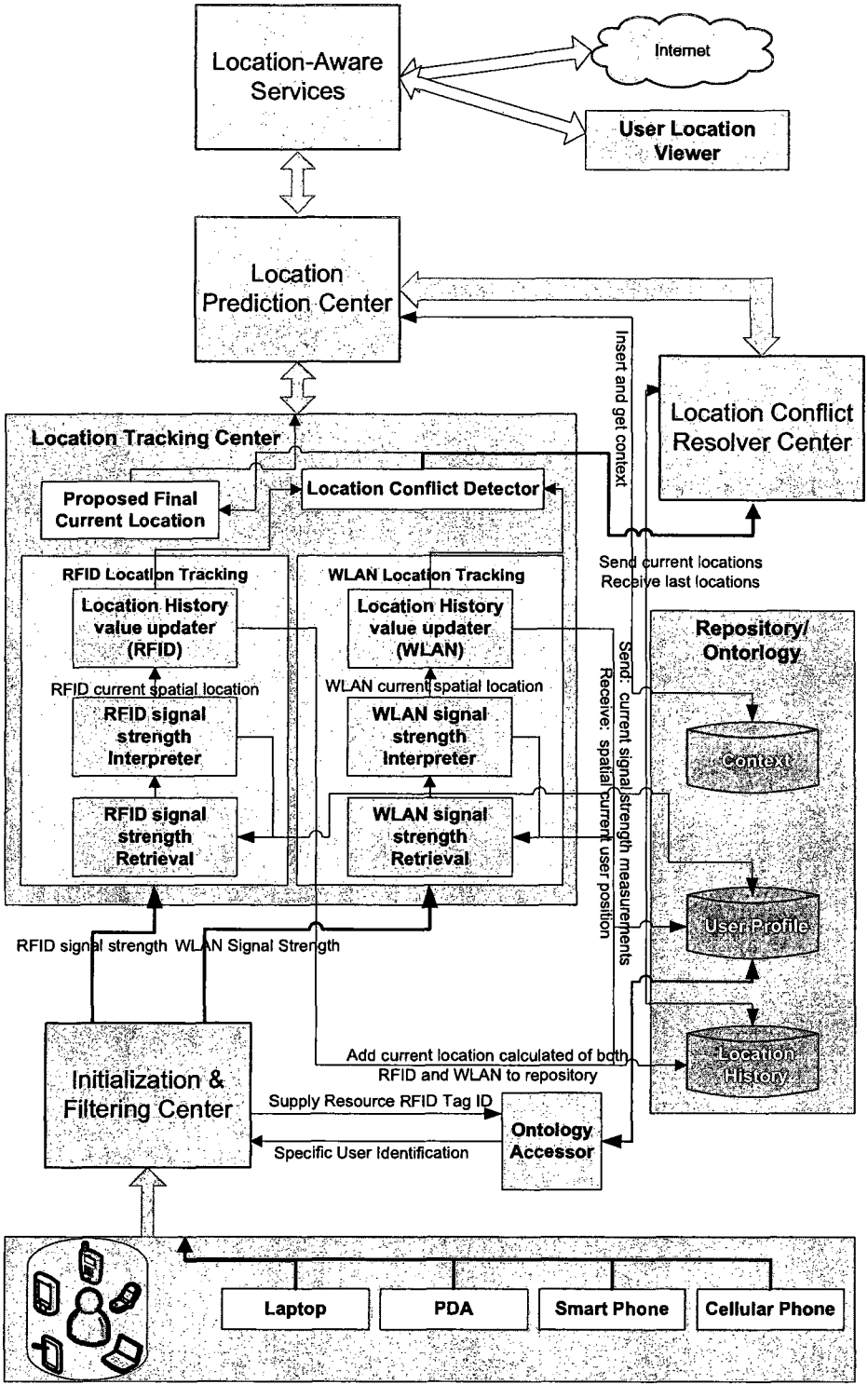



**Figure 4. 11. Integration of the Location Tracking Center with the system.**

### 4.2.3 Initialization and Filtering Center

This part of the system architecture runs on the users' mobile devices. It is responsible for initializing the process of location detection. It is made up of four classes: *User Acquisition Listener, WLAN Signal Calculation, SQL Database,* and *Packet Filterer.*

The program starts by listening to users entering the monitored area. The user is identified from the RFID packet, as shown in the previous section, and filtering the tag ID. The user's profile is extracted from the ontology and retrieved via the Ontology Accessor class to provide the system with all the available information concerning that specific user. This information consists of the user's interests, previous location history, specific profile information, and much more.

There are many ways of retrieving information concerning the signal strength received at the Ethernet card of a mobile computing device. We chose an efficient and organized way of retrieving signal strength information by using NetStumbler as mentioned previously. We extract the most recent information concerning signal strength from NetStumbler, store it in a SQL database to keep track of data extraction, and send it to the Location Tracking Center for current location calculation. Figure 4.12 illustrates how the Initialization and Filtering Center is integrated with the overall system structure.

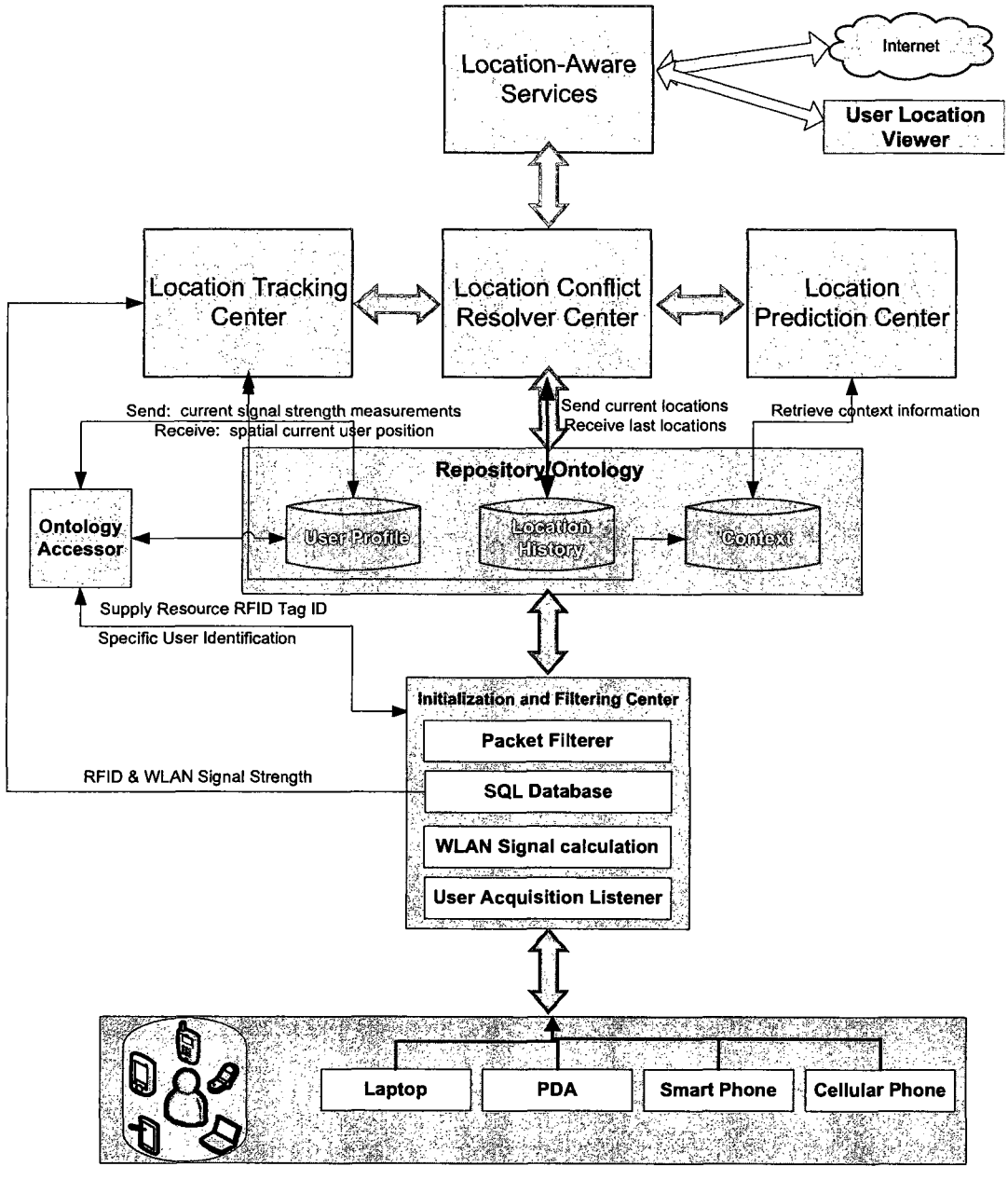



**Figure 4.12. Integration of the Initialization and Filtering Center with the system.**

### 4.2.4 Location Conflict Resolver Center

Now that we have two sets of locations provided by the Location Tracking Center that might or might not be the same, it is required to solve any conflict that may exist. Two conflict resolutions are proposed.

The first method is called "average location resolver." This method simply chooses the average of the two candidate locations without reference to locations stored in the history repository. For example, if the location proposed by the WLAN location estimation algorithm is CBY B502 and the location proposed by the RFID location estimation algorithm is CBY B504, the Average Location Resolver will choose CBY B503 as the final decision on the user's current location. Since location detection using the RFID method is more accurate than the WLAN method in estimating the current location of a user, it is desirable to assign a weighted average. Based on the evaluation results presented in Chapter Five, we see that RFID increases the accuracy of the system by an average of 13.83%. Therefore the weighted average will be as follows: 63.83% assigned to the RFID location estimation method, and 36.17% assigned to the WLAN location estimation method.

There exists a problem called "signal aliasing" which may affect locating the user. The signal strength at a point close to an AP may be similar to that at another point that is further away due to many obstacles; for example, the transmitted signal generally reaches the receiver through a multipath, meaning that the signal does not travel from the access point to the receiving device in a line of sight, rather it travels through the area by reflecting from one object to another until it reaches the destination device. Sometimes the number of people in the building may distort the signals travel and produce a distorted

version of the transmitted signal. Due to reflection, refraction, diffraction, and absorption of radio waves by structures and people inside a building, the WLAN Signal Strength map will not be accurate due to the fact that a map created at a particular time may not accurately reflect the environment at a different time.

As mentioned before, the solution to this problem is to use multiple WLAN Signal Strength maps to reflect different environmental states. But another solution is to use the second conflict resolving algorithm provided in our system, "Location History Resolver." The algorithm uses information about locations visited by the user in the past and stored in the ontology. The fact that a user's location at a given time is likely to be near another location at a previous time provides a solution. Therefore, the algorithm chooses the best candidate from data stored in the location history repository. The two candidate locations are compared with the user's previous locations, and the location closest to the previous locations is chosen. Figure 4.13 illustrates how the Location Tracking Center is integrated with the overall System structure.

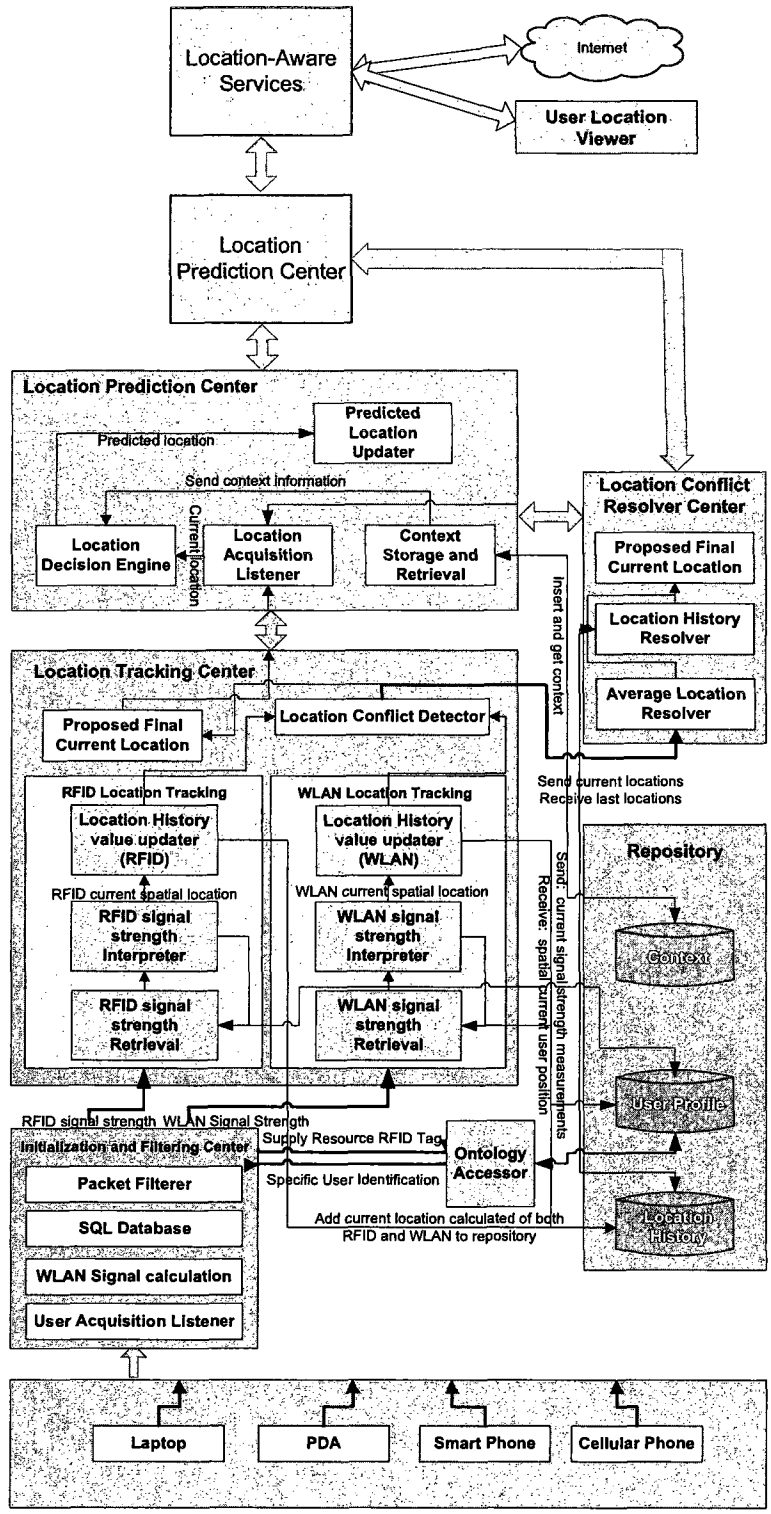



**Figure 4.13. Integration of the Location Conflict Resolver Center with the system.**

### 4.2.5 Location Prediction Center

With the current advances in the field of wireless technology, it is essential not just to locate a mobile user in real-time; predicting future locations a user will be heading towards is a key component in assisting many services. User mobility prediction represents a key component in modern ubiquitous computing environments. Mobility prediction will be a major source of assistance in handoff management, resource reservation, and service pre-configuration, achieving a seamless transfer or delivery in services. Fast and accurate mobility prediction techniques have become one of the main topics in current research efforts. The importance of mobility prediction techniques can be seen at both the network and service levels.

As mentioned previously, the prediction method relies on the work conducted by Nancy Samaan [8] using the Dempster-Shafer Theory [18]. With the cooperation of Moyad Aloqaily [39], we were able to integrate the prediction algorithm within our system architecture. Once a user's current location has been identified, future location is predicted in order to provide better services. Many attempts have been made to address the issue of prediction in mobility [8], [16], [17], [19]. Our proposal is unique because it makes use of a user's "current-location identification" as described in the system architecture. After an output from the location tracking center is retrieved by the prediction center, context and history locations are both queried from the context and location history repository, respectively. The Dempster-Shafer Theory uses these as inputs to its algorithm. A weight is given to each future candidate location, and the one with the highest weight is chosen. The successful approach in dealing with problems of combining different bodies of evidence to reach a decision in situations with a high

degree of uncertainty has attracted considerable attention to this mathematical theory. A detailed explanation of the prediction algorithm is provided in Appendix A. The Location Prediction Center is composed of 3 classes: *Context Storage and Retrieval Class*, *Location Acquisition Listener Class*, and *Mobility Prediction Class*. The listener retrieves current locations estimated by the Location Tracking Center or locations corrected by the Location Resolver Center. Context Storage and Retrieval class acquires users' schedules, goals, tasks, and other information concerning users required by the Dempster-Shafer Theory from the repository to predict future candidate locations and decide on a final one using the Mobility Prediction Class. Figure 4.14 shows the Location Prediction Center.

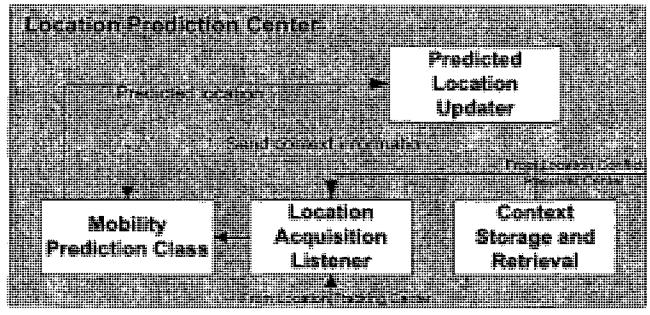



**Figure 4.14. Location Prediction Center.**

### 4.2.6 Overview of the System Architecture

We discussed in the previous sections each component of the system architecture and discussed details. As an overview of the system, a recap of the components will be looked at. The system is divided into the following main parts: Repository, Location Tracking Center, Location Conflict Resolver Center, and Location Prediction Center. Figure 4.15 shows the overall system architecture integrated with the Location Prediction Center. The Repository uses ontology-based context storage; its unified method of modeling and

representation is flexible and extensible enough to handle both a wide range of context types and the relationships between them. Current location is identified by both WLAN and RFID technology. The signal strength emitted from both WLAN access points and RFID readers is obtained and processed separately through the Location Tracking Center. Conflicts resulting from the WLAN and RFID location identification algorithms are resolved through the Location Conflict Resolver Center using two algorithms: Average Location and Location History. Once a user's current location has been identified, future location is predicted through the Location Prediction Center. This project can be further developed to be integrated with other applications either through the same network or other wider networks, such as the Internet. Applications include services such as displaying a map of the surroundings to guide the user inside a building or printing a document on the nearest printer. Chapter 5 will provide an explanation of simple service developed in this project by providing a GUI monitoring application that shows the location of users in real time on a map of the University of Ottawa campus.

## 4.3 Summary

The location-aware system architecture presented within this chapter described an alternative solution to current architectures. The main components of the system were explained clearly in details earlier. Those components are: Initialization and Filtering Center, Location Tracking Center, and Location Conflict Resolver Center.

Location tracking was carried out using to sets of technologies: WLAN and RFID, with the integration of the ontology. We presented a user mobility prediction scheme that predicts the travelling destination using a user's current location, preferences, goals and

schedules applied to a mathematical theory of evidence. Background information concerning the Dempster-Shafer Theory is introduced in Appendix A to help understand how the Location Prediction Center applies the theory to predict future locations visited by users.

To illustrate the functionality of our proposed location-aware system architecture, we have built a prototype system that will be illustrated in Chapter Five.

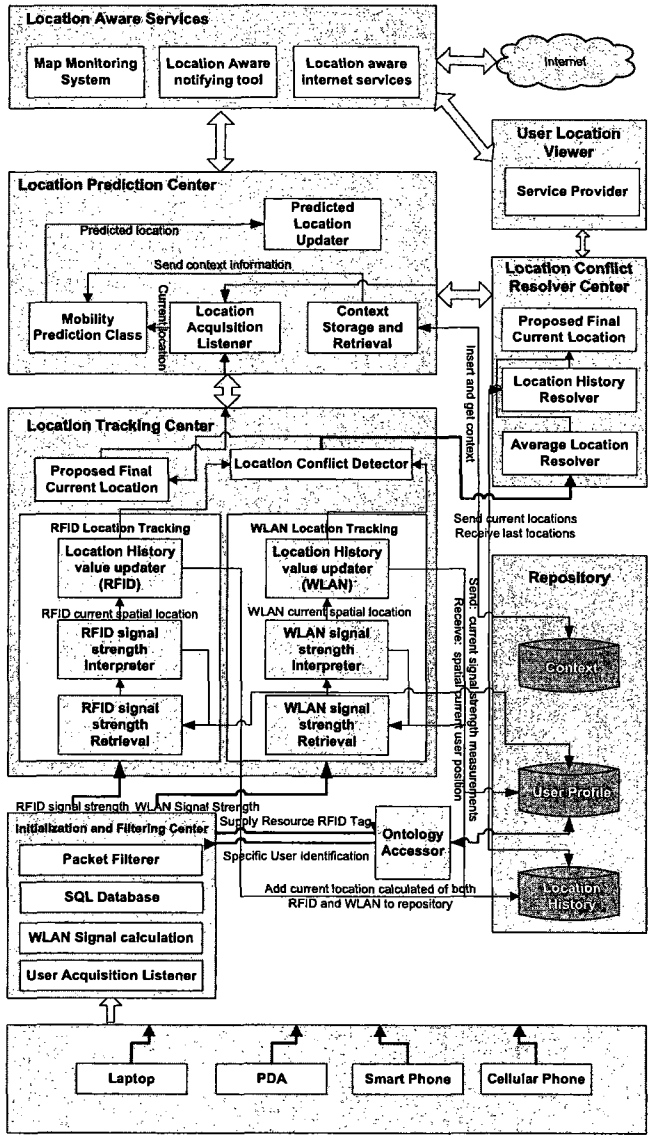



**Figure 4.15. Overall System Architecture.**

# Chapter 5

# Prototype Implementation and Evaluation

## 5.1 Objective

This chapter presents a prototype implementation of our proposed location-aware system architecture, which utilizes tracking a user in a university campus, more precisely in the Colonel By building at the University of Ottawa. We have managed to implement all components and features of our architecture, with the exception of the Location Prediction Center. This feature has been replaced with results gathered by simulations from [8] in our Lab.

The system was first implemented based on a two-stage process. In the first stage, the Ontology was developed using Protégé OWL [38]. The ontology, as presented in Chapter Three, represented a general model through which context knowledge within all domains can be modeled directly or by extending the Ontology. The second step involved programming the system architecture using the Java programming language.

The system was then evaluated to verify the positioning system's practicality and show the experimental result. The positioning system was investigated using some experiments and compared with other existing systems to reveal the strengths and weaknesses of our system.

## 5.2 Scenario

To demonstrate the abilities of the system, the following scenario was conducted at the University of Ottawa (Figure 5. 1). A student entering a building on campus is checked into the coverage area of an RFID reader (step 1). The student's identification number is sent to the system, and the user profile is retrieved (step 2). The wireless interface card gathers information on signal strength of beacon packets emitted from different access points (step 3). Those values are sent to the location tracking center (step 4). A location is estimated and stored in the location history repository (step 5). The process of location estimation using the RFID data estimates a location that is stored in the location history repository. If a conflict is detected, it is resolved by the "location conflict resolver" (step 6). The steps are repeated dynamically to determine the student's current location. Because the student is known to be walking down a specific hall, the system predicts that he or she is heading towards his or her office (step 7).

While the student is in the hall, he or she is reminded of meetings, schedules, appointments, and video messages on her mobile device. At the same time, a session handoff [7] process from the mobile device to the student's desktop is initiated and put on hold until he or she enters the office. Once he or she is inside the office, all programs running are transferred to his or her desktop, and he or she is able to continue his or her work from there. This process is repeated during the day, with context information used to start specific services at a specific time.

The steps in Figure 5. 1 can also be applied to a sequence diagram. A simplified example is shown in Figure 5. 2.

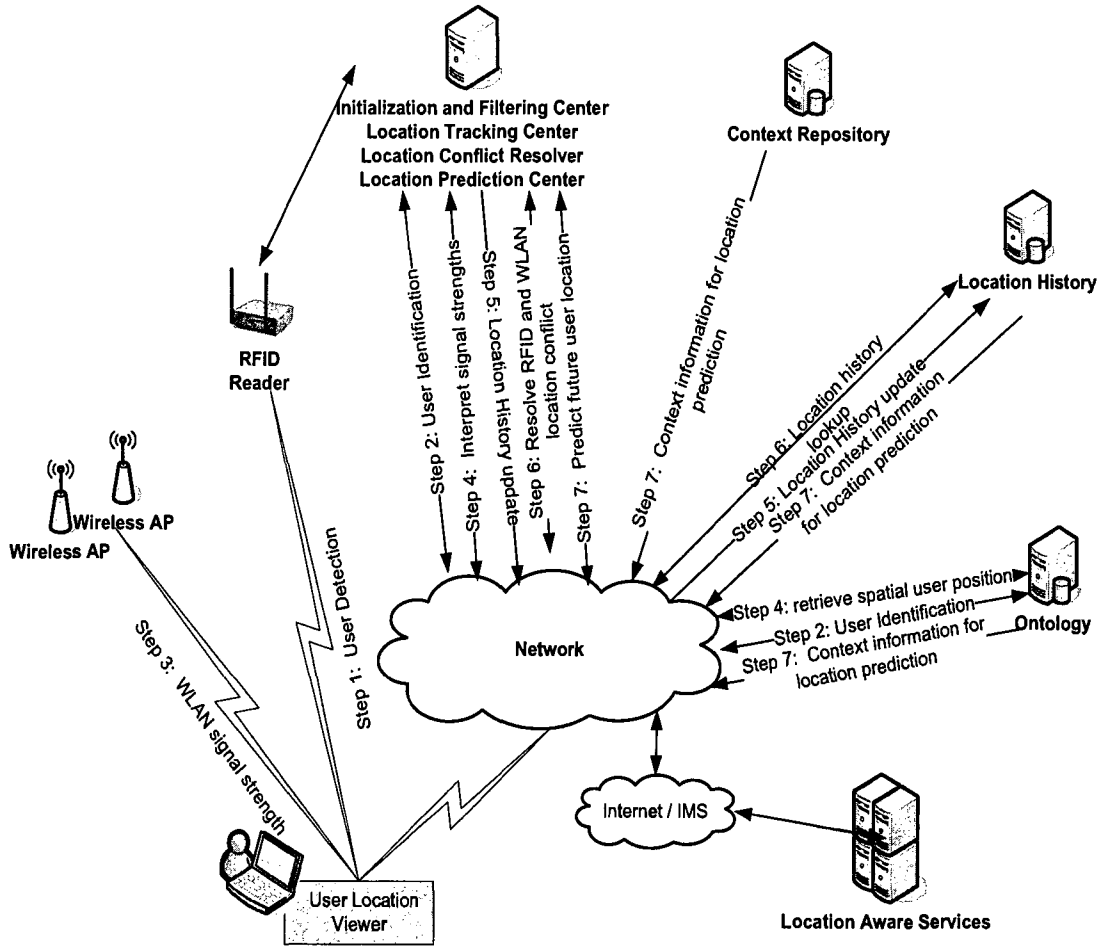



**Figure 5. 1. Steps involved in locating the user, and predicting their future location.**

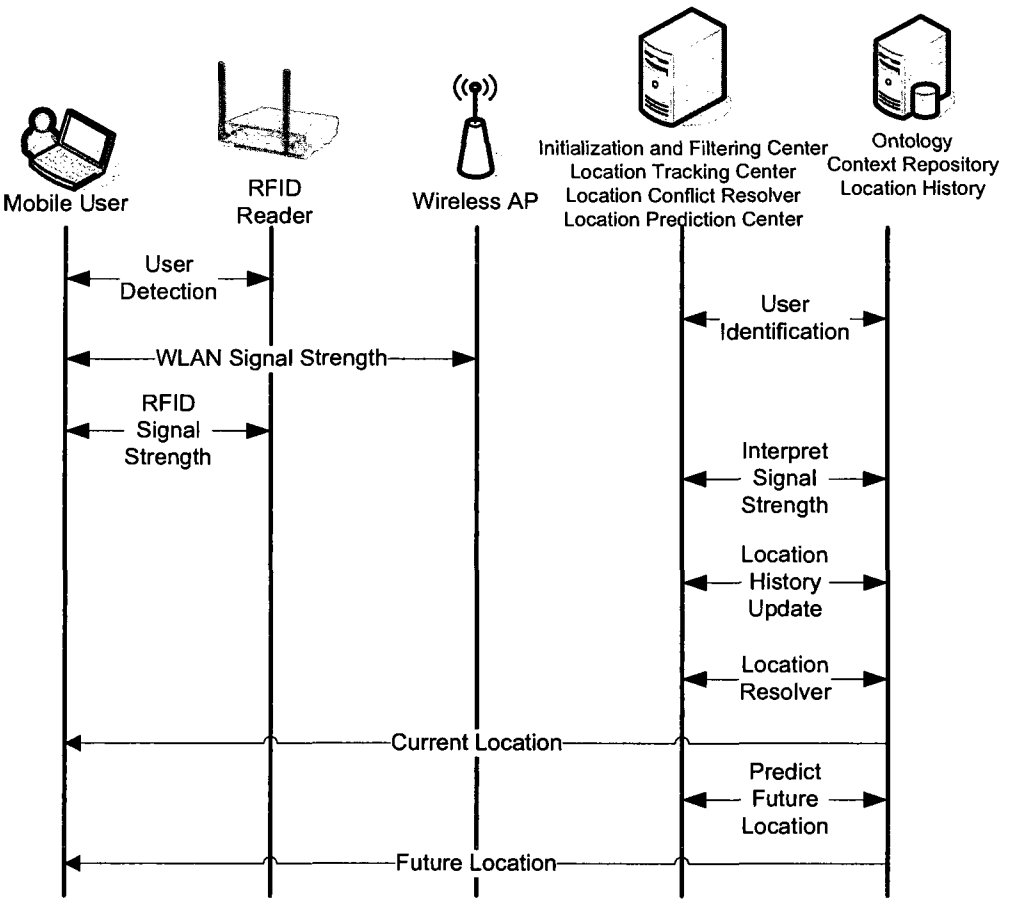



**Figure 5. 2. Sequence diagram of location detection and prediction.**

## 5.3 Prototype Implementation

Based on the analysis of the previous chapters, we developed a prototype for the University of Ottawa in order to evaluate the architecture.

Protégé [38] was used to model the ontology which represented a model through which context knowledge within all domains can be modeled directly as well as an extension to our domain specific context. Figure 5.3 provides an overview of the environment in which context is modeled.

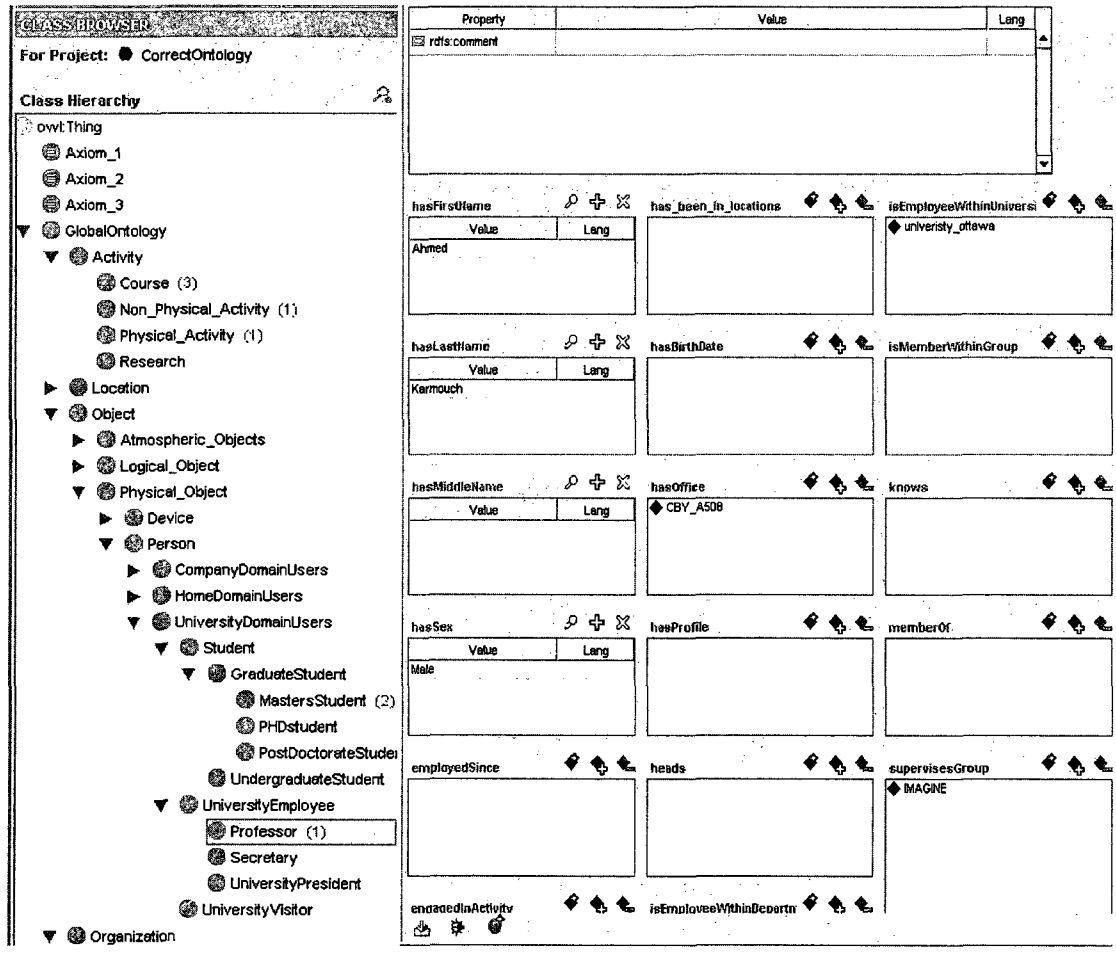



**Figure 5.3. Context modeling environment.**

The following subsections illustrates how the five classes of the ontology are modeled using the OWL-based language demonstrated through examples.

### *5.3.1 Location Example*

Illustrated in Figure 5.4 is an example of how location is modeled using OWL within the ontology. The *GeographicalPosition* and *SpatialProximity* classes and their subclasses are presented by modeling a research laboratory in room B502 on the 5$^{th}$ floor of the Colonel By Building located on 161 Louis Pasteur Street, in the city of Ottawa, Ontario, Canada.

From the example we can see that a relationship is established between the room and an instance of the *Floor* class through the *isRoomWithinFloor* property. A *SpatialProximity* relationship exists between the 5$^{th}$ floor and the 4$^{th}$ floor using an instance of the *Above* class. The *isFloorWithinBuilding* is used to represent its address by using the *GeographicalAddress* class, indicating that the room is located on 161 Louis Pasteur Street, in the city of Ottawa, in the Canadian province of Ontario.

```
<Room rdf:ID="Room_B502">
  <hasRoomNumber rdf:datatype="&xsd;string">B502</hasRoomNumber>
  <hasRoomType rdf:datatype="&xsd;string"
  >Research Laboratory</hasRoomType>
  <isRoomWithinFloor rdf:resource="#Fifth_Floor"/>
</Room>
<Floor rdf:ID="Fifth_Floor">
  <hasFloorNumber rdf:datatype="&xsd;int">5</hasFloorNumber>
  <isFloorWithinBuilding rdf:resource="#ColonelBy"/>
  <hasComparativeLocation rdf:resource="#Above_1"/>
</Floor>
<Above rdf:ID="Above_1">
  <inComparisonTo rdf:resource="#Fourth_Floor"/>
</Above>
<Floor rdf:ID="Fourth_Floor">
  <hasFloorNumber rdf:datatype="&xsd;int">4</hasFloorNumber>
  <isFloorWithinBuilding rdf:resource="#ColonelBy"/>
</Floor>
<Building rdf:ID="ColonelBy">
  <hasBuildingType rdf:datatype="&xsd;string"
    >Engineering Building</hasBuildingType>
  <hasBuildingName rdf:datatype="&xsd;string">Colonel By</hasBuildingName>
  <hasBuildingNumber rdf:datatype="&xsd;string">161</hasBuildingNumber>
  <hasAddress rdf:resource="#CBY_Address"/>
  <buildingHasRoom rdf:resource="#Room_B502"/>
</Building>
<GeographicalAddress rdf:ID="CBY_Address">
  <hasPostalCode rdf:datatype="&xsd;string">K1N 6N5</hasPostalCode>
  <hasCity rdf:resource="#Ottawa_City"/>
  <hasStreet rdf:resource="#Louis_Street"/>
  <hasCountry rdf:resource="#Canada"/>
  <hasProvince_State rdf:resource="#Ontario_Province"/>
</GeographicalAddress>
<Street rdf:ID="Louis_Street">
  <hasStreetName rdf:datatype="&xsd;string">Louis Pasteur</hasStreetName>
  <isWithinCity rdf:resource="#Ottawa_City"/>
</Street>
<City rdf:ID="Ottawa_City">
  <hasCityName rdf:datatype="&xsd;string">Ottawa</hasCityName>
  <isWithinProvince_State rdf:resource="#Province_State"/>
</City>
<Province_State rdf:ID="Ontario_Province">
  <hasProvince_StateName    rdf:datatype="&xsd;string">Ontario</hasProvince_StateName>
  <isWithinCountry rdf:resource="#Country"/>
</Province_State>
<Country rdf:ID="Canada">
  <hasCountryName rdf:datatype="&xsd;string">Canada</hasCountryName>
</Country>
```

**Figure 5.4. Example of location ontology modeled in OWL.**

### 5.3.2 *Time Example*

To illustrate the time ontology, an example is provided in Figure 5.5. The example illustrates how exact time and comparative time are modeled using OWL within the ontology. A scheduled meeting is set to occur after the user's daily exercises that occur at 8:15:00 a.m. EST. This is modeled using the *hasStartTime* relationship to the *Activity* class. If the exercise's end time is not known, another non-physical Activity (Meeting 36) uses comparative time to declare that its lecture would start 'after' the daily exercise.

```
<Physical_Activity rdf:ID="Daily_Excersices">
<hasStartTime rdf:resource="#Daily_Excersices_StartHour"/>
<hasStartTime rdf:resource="#Daily_Excersises_StartMinute"/>
<hasStartTime rdf:resource="#ceg3180_StartSecond"/>
</Physical_Activity>
<Non_Physical_Activity rdf:ID="Meeting_36">
<hasStartTime rdf:resource="#After_Excersises"/>
</Non_Physical_Activity>
<After rdf:ID="After_Excersises">
<inRelationToActivity rdf:resource="#Daily_Excersices"/>
</After>
<Hour rdf:ID="Daily_Excersices_StartHour">
<timeZone rdf:datatype="&xsd;string">EST</timeZone>
<am_pm_24 rdf:datatype="&xsd;string">pm</am_pm_24>
<hasHourValue rdf:datatype="&xsd;int">8</hasHourValue>
</Hour>
<Minute rdf:ID="Daily_Excersises_StartMinute">
<hasMinuteValue rdf:datatype="&xsd;int">15</hasMinuteValue>
</Minute>
<Second rdf:ID="Daily_Excersises_StartSecond">
<hasSecondValue rdf:datatype="&xsd;int">0</hasSecondValue>
</Second>
```

**Figure 5.5. Example of time ontology modeled in OWL.**

### *5.3.3 Object Example*

The following example shown in Figure 5. 6 illustrates how objects are modeled using OWL within the ontology. An active RFID tag within the RFID group "GRFCM11" belongs to Ismaeel Al Ridhawi, who is a student within the Master's program. The existence of this user requires the creation of an instance of the *MastersStudent* class such as the "*Ismaeel*" instance provided here. The example provides a description of a Master's student within the University of Ottawa, displaying the student's full name, thesis title, year of study, and relationships to other individuals. The example shows that Ismaeel has an office located in "CBY_A508."

Also, an instance was created for the *RFID_Tag* class such as the "Tag_04568715" instance provided in the example. A description of the tag ID, the RFID group that it belongs to, and the person that it represents is shown in the figure. The figure also shows the representation of access points and RSS values in the ontology.

```
<RFID_Tag rdf:ID="Tag_04568715">
    <hasRFIDtagID rdf:datatype="&xsd;int">4568715</hasRFIDtagID>
    <belongsTo rdf:resource="#Ismaeel"/>
    <hasRFIDgroup rdf:resource="#GRFCM11"/>
  </RFID_Tag>
RFID_Reader rdf:ID="GRFCM11">
    <hasRFIDgroupID rdf:datatype="&xsd;string">GRFCM11</hasRFIDgroupID>
  </RFID_Reader>
<MastersStudent rdf:ID="Ismaeel">
    <hasFirstName rdf:datatype="&xsd;string">Ismaeel</hasFirstName>
    <hasStudentNumber rdf:datatype="&xsd;int">3385278</hasStudentNumber>
    <hasSex rdf:datatype="&xsd;string">Male</hasSex>
    <hasMiddleName rdf:datatype="&xsd;string">Kadhim</hasMiddleName>
    <MscThesisTitle rdf:datatype="&xsd;string"
      >Location Tracking System</MscThesisTitle>
    <hasLastName rdf:datatype="&xsd;string">Al Ridhawi</hasLastName>
    <yearOfStudy rdf:datatype="&xsd;int">2</yearOfStudy>
    <isStudentWithinDepartment rdf:resource="#electrical_engineering"/>
    <hasOffice rdf:resource="#CBY_A508"/>
    <hasBirthDate rdf:resource="#ismaeel_birth"/>
    <knows rdf:resource="#Yousif"/>
    <isStudentWtihinFaculty rdf:resource="#graduate_faculty"/>
    <hasSupervisor rdf:resource="#Karmouch"/>
    <isStudentWtihinUniversity rdf:resource="#univeristy_ottawa"/>
    <isMemberWithinGroup rdf:resource="#IMAGINE"/>
  </MastersStudent>
<Signal rdf:ID="B502_signal">
    <hasWIFISignalGroups rdf:datatype="&xsd;int">2</hasWIFISignalGroups>
    <signalType rdf:datatype="&xsd;string">WIFI</signalType>
    <belongsToRoom rdf:resource="#CBY_B502"/>
    <belongsToAccessPoint rdf:resource="#B502_MPRL802.11G-81"/>
    <belongsToAccessPoint rdf:resource="#Window_liksys-80"/>
  </Signal>
<AccessPoint rdf:ID="B502_MPRL802.11G-81">
    <hasDeviceName rdf:datatype="&xsd;string">MPRL802.11G</hasDeviceName>
    <hasMacAddress rdf:datatype="&xsd;string"
      >00:13:46:fa:79:20</hasMacAddress>
    <hasModel rdf:datatype="&xsd;string">0523</hasModel>
    <hasSignalStrength rdf:datatype="&xsd;int">81</hasSignalStrength>
    <hasLocation rdf:resource="#CBY_B502"/>
  </AccessPoint>
```

**Figure 5. 6. Example of object ontology modeled in OWL.**

### *5.3.4 Organization Example*

The following simplified example, shown in Figure 5.7, illustrates how organization is modeled using OWL within the ontology. The ontology describes the relationship between the University of Ottawa, the Faculty of Graduate and Postgraduate Studies, and the Department of Electrical Engineering.

```
<University rdf:ID="univeristy_ottawa">
  <hasName rdf:datatype="&xsd;string">The University Of
  Ottawa</hasName>
  <hasFaculty rdf:resource="#graduate_faculty"/>
</University>
<Faculty rdf:ID="graduate_faculty">
  <hasName rdf:datatype="&xsd;string">Faculty Of Graduate and
  Postgraduate Studies</hasName>
  <isFacultyWithinUniversity rdf:resource="#univeristy_ottawa"/>
  <hasDepartment rdf:resource="#electrical_engineering"/>
</Faculty>
```

**Figure 5.7. Example of organization ontology modeled in OWL.**

### *5.3.5 Activity Example*

The following simplified example, shown in Figure 5.8, illustrates the use of ontology to model Activity. The ontology describes the Daily Exercises conducted by a user and the relationship between the Activity class and Time. Daily Exercises have a specific start time, as illustrated in the figure.

```
<Physical_Activity rdf:ID="Daily_Excersices">
    <hasStartTime rdf:resource="#Daily_Excersises_StartMinute"/>
    <hasStartTime rdf:resource="#Daily_Excersices_StartHour"/>
    <hasStartTime rdf:resource="#Daily_Excersises_StartSecond"/>
  </Physical_Activity>
<Hour rdf:ID="Daily_Excersices_StartHour">
    <timeZone rdf:datatype="&xsd;string">EST</timeZone>
    <am_pm_24 rdf:datatype="&xsd;string">pm</am_pm_24>
    <hasHourValue rdf:datatype="&xsd;int">8</hasHourValue>
  </Hour>
  <Minute rdf:ID="Daily_Excersises_StartMinute">
    <hasMinuteValue rdf:datatype="&xsd;int">15</hasMinuteValue>
  </Minute>
  <Second rdf:ID="Daily_Excersises_StartSecond">
    <hasSecondValue rdf:datatype="&xsd;int">0</hasSecondValue>
  </Second>
```

**Figure 5.8. Example of activity ontology modeled in OWL.**

## 5.4 Implementation of Architecture Components

Java programming language was used for the "Initialization and Filtering Center," "Location Tracking Center," and "Location Conflict Resolver." They are stored in servers running on 2.00 GB, 3.6GHz Intel Pentium D computers. Thin clients run on a 2.0 GHz Intel Pentium Centrino laptop with an RF Code M100 active RFID tag installed and using NetStumbler to gather information on the available access points and the Received Signal Strength (RSS) values.

During the information gathering process, a text file is created concerning the signal strength and values are stored into the file periodically every second as shown in Figure 5.9. The most recent RSS values are extracted from the text file, sent to the Initialization and filtering center at the server, and stored in a MySQL Database as shown in Figure 5.10. RFID values are extracted in the same manner and are stored in a temporary MySQL Database as well.

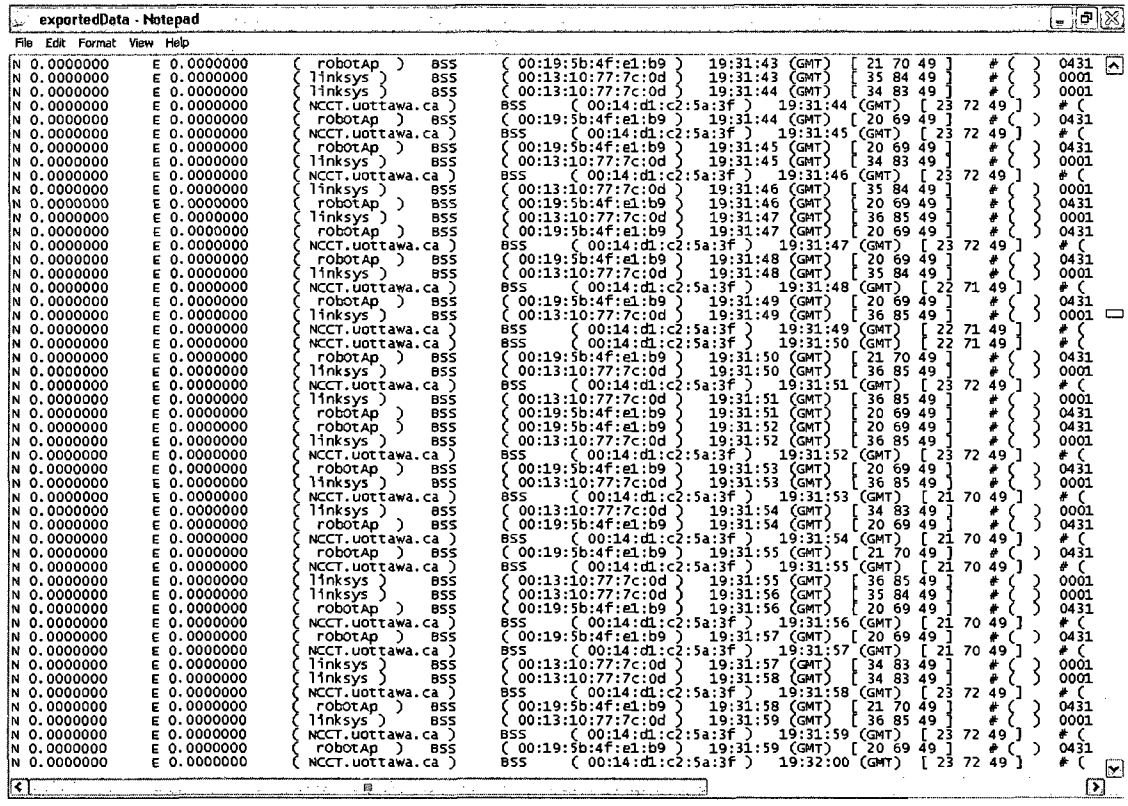



**Figure 5.9. Data extracted from Netstumbler.**

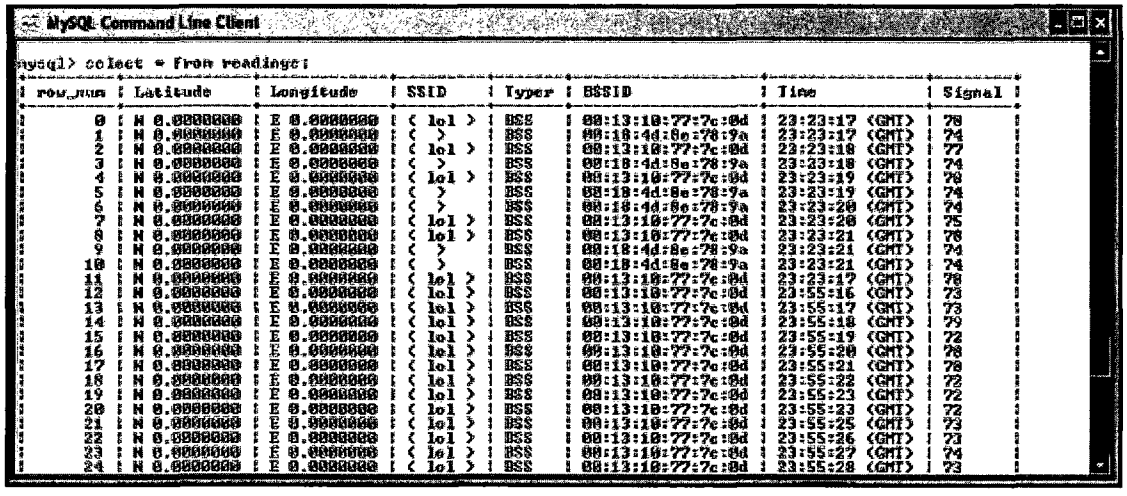



**Figure 5.10. WLAN RSS value data stored in Database.**

To begin the negotiation session in which the client sends the RSS values and the server sends back the location, the user must first establish a connection to the server. Figure 5.11 shows the server side before connection establishment. Once the connection has been established by running the user client program installed on the client's mobile device, NetStumbler will run in the background and collect RSS information which is sent to the server, where it is stored in the SQL database. Figure 5.12 and Figure 5.13 show the server and client with a connection established between them.

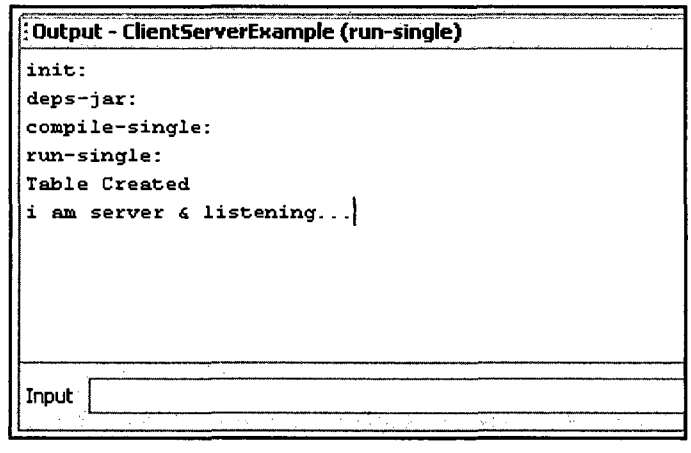



**Figure 5.11. Server before connection establishment.**

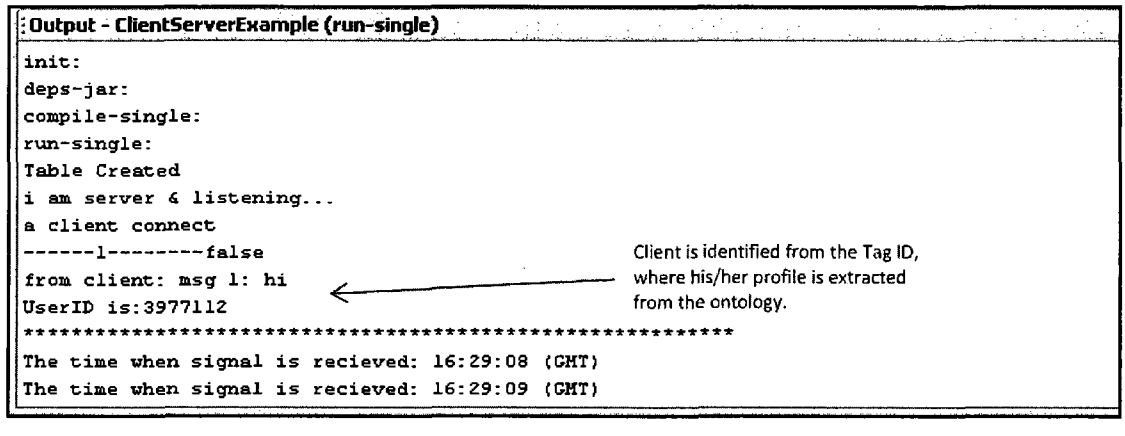



**Figure 5.12. Connection established between client and server (server side).**

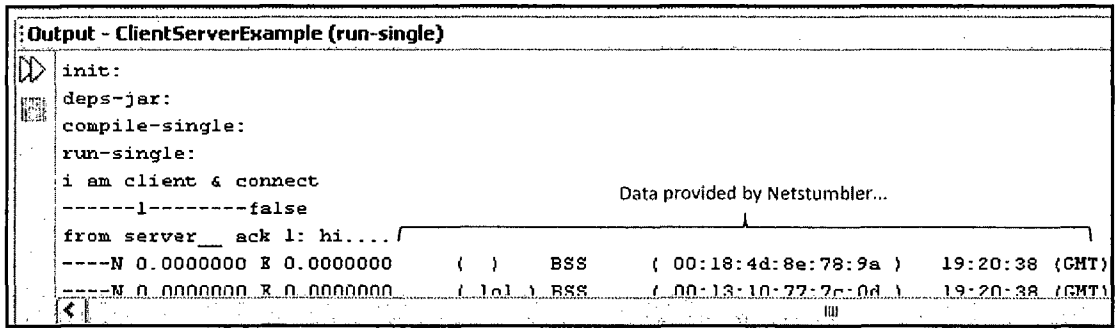
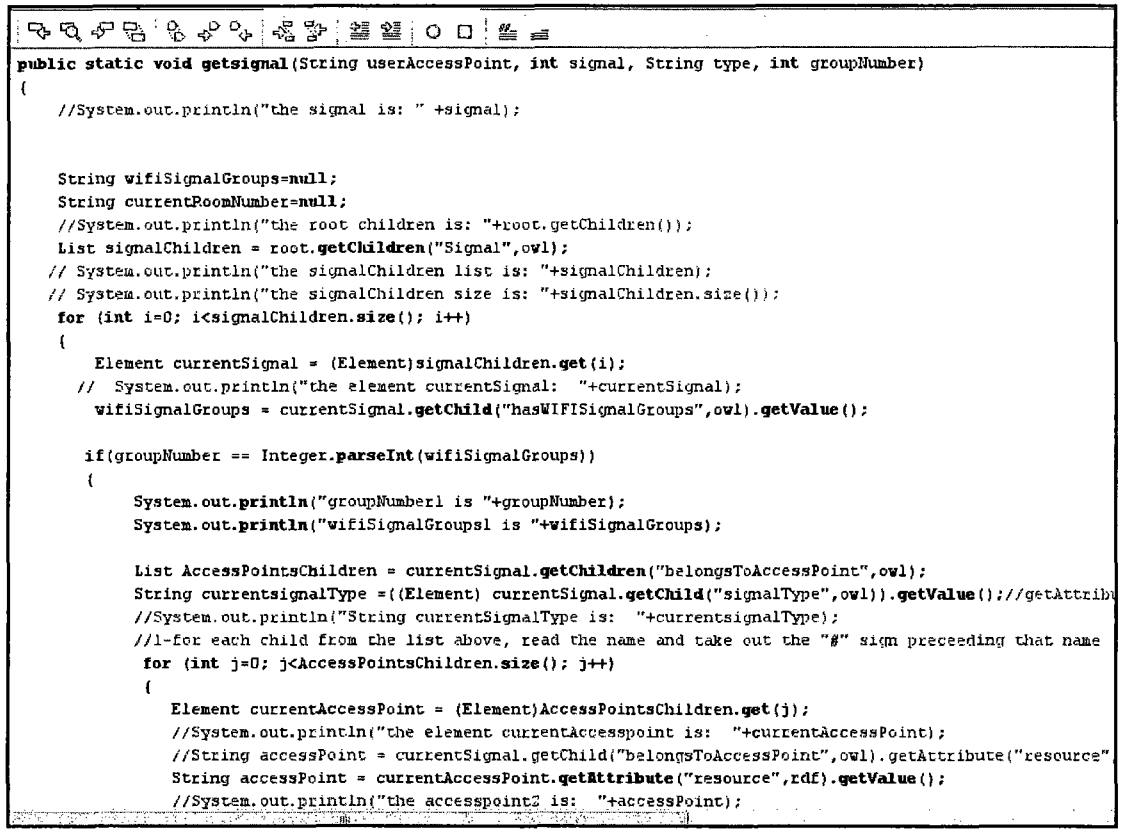



**Figure 5.13. Connection established between client and server (client side).**

In Figure 5.14, we provide a snapshot of the partial code in Java used to extract the profile of a user from the Ontology. Also Figure 5.15, provides a snapshot of the Java code of the algorithm used to locate the user using signal strength values of WLAN and RFID.

```
public static void getUserProfile(String tagID2)
{
    Element currentChild=null;
    List tagChildren = root.getChildren("RFID_Tag",owl);
    for (int i=0; i<tagChildren.size(); i++)
    {
        Element currentTag=(Element)tagChildren.get(i);
        String tagID=currentTag.getChild("hasRFIDtagID",owl).getValue();
        //System.out.println(tagID);
        if (tagID.equals(tagID2))
        {
            String ownerInstance=((Element)currentTag.getChild("belongsTo",owl)).getAttribute("resource",rdf).
            //System.out.println(ownerInstance);
            //List tagProfileChildren = root.getChildren("RFID_Tag",owl);
            List allChildren=root.getChildren();
            for(int j=0; j<allChildren.size();j++)
            {
                //System.out.println("current J: "+j);
                currentChild=(Element) allChildren.get(j);
                //System.out.println(currentChild.getName());
                if(!currentChild.getName().equalsIgnoreCase("Ontology")){
                    if(currentChild.getAttributeValue("ID",rdf).equalsIgnoreCase(ownerInstance.substring(1))){
                    break;
                    }
                }
            }
```

**Figure 5.14. Java code of Profile extraction from Ontology.**



```
public static void getsignal(String userAccessPoint, int signal, String type, int groupNumber)
{
    //System.out.println("the signal is: " +signal);


    String wifiSignalGroups=null;
    String currentRoomNumber=null;
    //System.out.println("the root children is: "+root.getChildren());
    List signalChildren = root.getChildren("Signal",owl);
   // System.out.println("the signalChildren list is: "+signalChildren);
   // System.out.println("the signalChildren size is: "+signalChildren.size());
    for (int i=0; i<signalChildren.size(); i++)
    {
        Element currentSignal = (Element)signalChildren.get(i);
      //  System.out.println("the element currentSignal:  "+currentSignal);
       wifiSignalGroups = currentSignal.getChild("hasWIFISignalGroups",owl).getValue();

      if(groupNumber == Integer.parseInt(wifiSignalGroups))
      {
           System.out.println("groupNumber1 is "+groupNumber);
           System.out.println("wifiSignalGroups1 is "+wifiSignalGroups);

           List AccessPointsChildren = currentSignal.getChildren("belongsToAccessPoint",owl);
           String currentsignalType =((Element) currentSignal.getChild("signalType",owl)).getValue();//getAttrib
           //System.out.println("String currentSignalType is:  "+currentsignalType);
           //1-for each child from the list above, read the name and take out the "#" sign preceeding that name
            for (int j=0; j<AccessPointsChildren.size(); j++)
            {
               Element currentAccessPoint = (Element)AccessPointsChildren.get(j);
               //System.out.println("the element currentAccesspoint is:  "+currentAccessPoint);
               //String accessPoint = currentSignal.getChild("belongsToAccessPoint",owl).getAttribute("resource"
               String accessPoint = currentAccessPoint.getAttribute("resource",rdf).getValue();
               //System.out.println("the accesspoint2 is:  "+accessPoint);
```

**Figure 5.15. Java Code of location tracking algothim.**

Once a location has been identified for a user by the server, the client is notified of the room he is currently in. Figure 5.16 and Figure 5.17 provide a snapshot of both the server and client with the location identified.

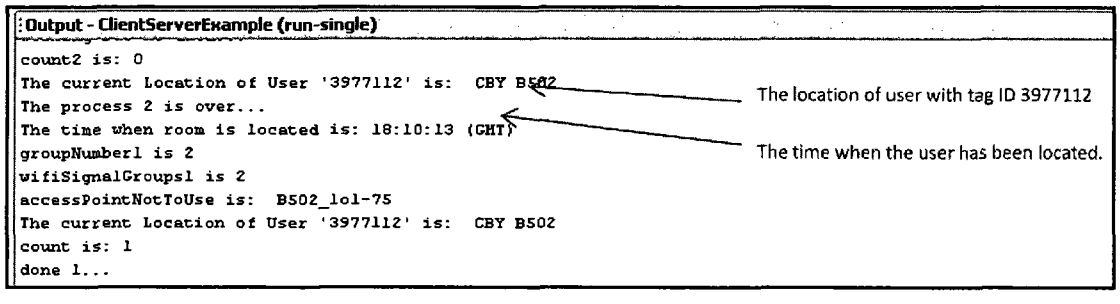



**Figure 5.16. Notification of Location at the server.**

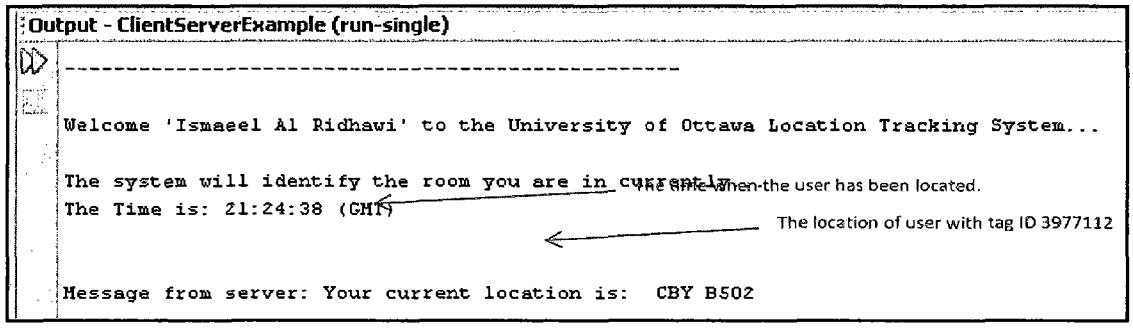



**Figure 5.17. Notification of Location at the client.**

We developed a GUI monitoring application that shows the location of users in real time in red on the map of the campus. Reminders and important information about the user can be displayed inside the white box on the graphical user interface. Figure 5.18 shows a screenshot of the interface.

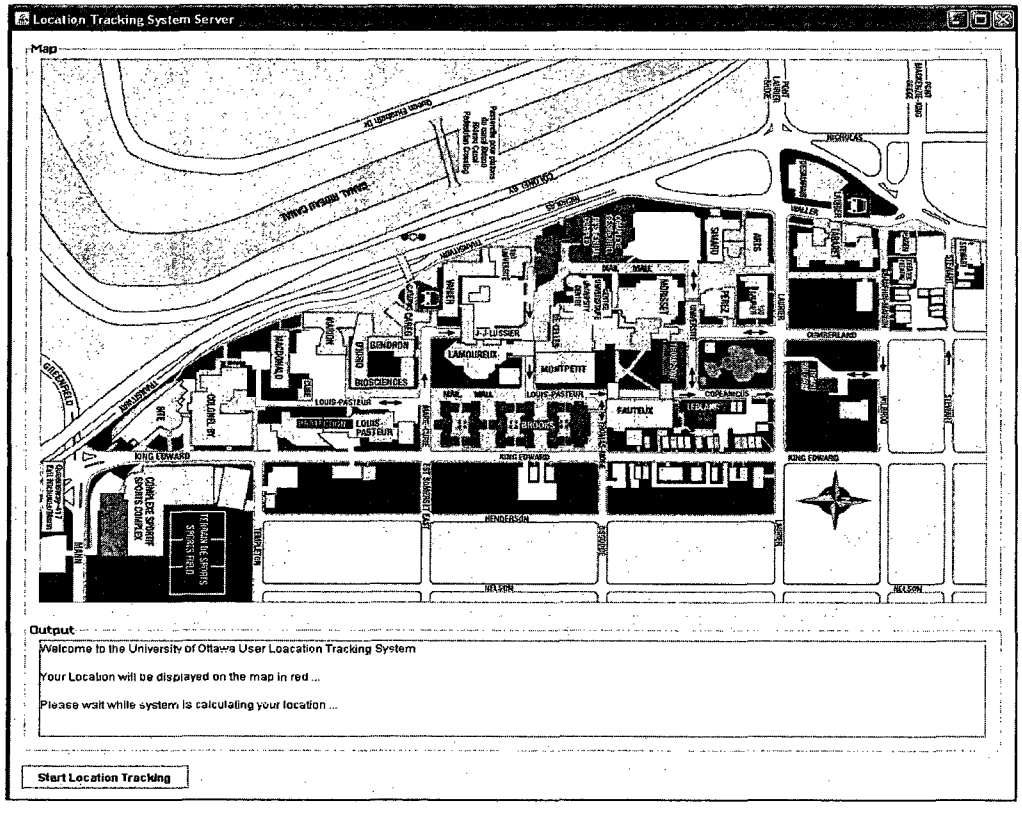



**Figure 5.18. User Interface of a client window.**

## 5.5 Evaluation

The evaluation was based on the scenario described above and conducted on the fifth floor of the Colonel By Building at the University of Ottawa (Figure 5.19). Data such as signal strength, user arrival and departure times, places of interest and destinations was collected over a couple of months for evaluation and to allow continual improvements to the location learning and predicting algorithm. It is important to note that the data we have collected was very time consuming; and due to time and area constraints, we were unable to collect as much data necessary in order to come up with a strong conclusion.

The wide availability of access points in the building contributes to the accuracy of the system. There are up to 10 access points in the experimental area but this alone is not enough for the system to be accurate. The installation of RF Code M200 fixed RFID readers (Figure 5.20) in the building has greatly assisted our ability to calculate user location separately from WLAN measurements. The increase in accuracy is clearly seen in on-site tests.

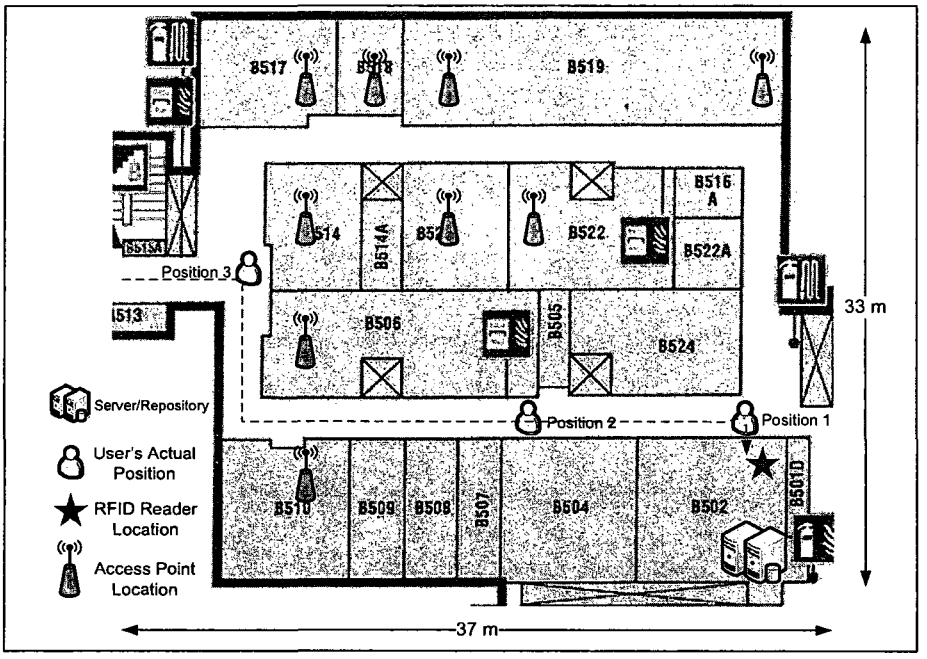



**Figure 5.19. System evaluation area.**

To determine the consistency of signal strength over time, samples were taken at five-second intervals over a period of 48 hours. The results (Figure 5.21) show that signal strength variation is fairly stable at +/- 5 dBm.

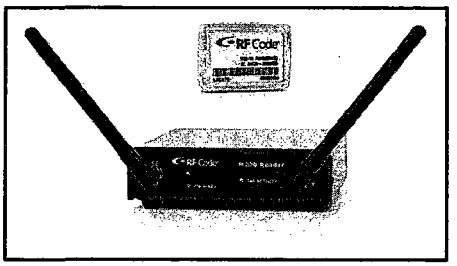



**Figure 5.20. RF Code M200 fixed RFID reader and M100 active tag.**

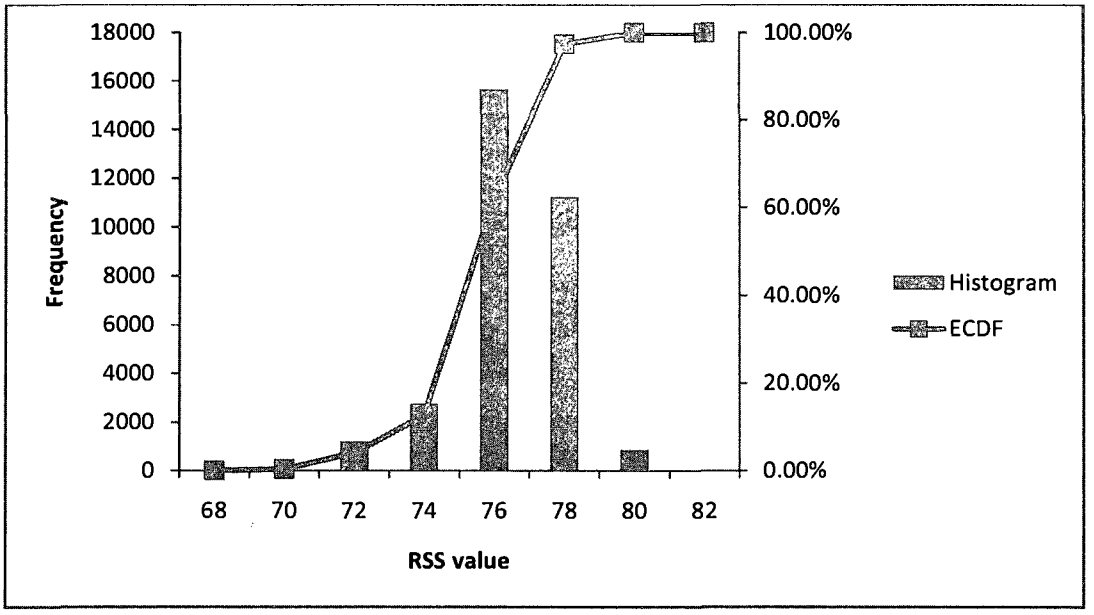



**Figure 5.21. RSS measurements over time.**

The RSS in the environment was also measured; the results are shown in Figure 5.22. The results show that RSS decreases with distance and when obstacles obscure the line of sight between the mobile device and the access point. This information led to the use of RFID technology and the enhancement of the algorithm.

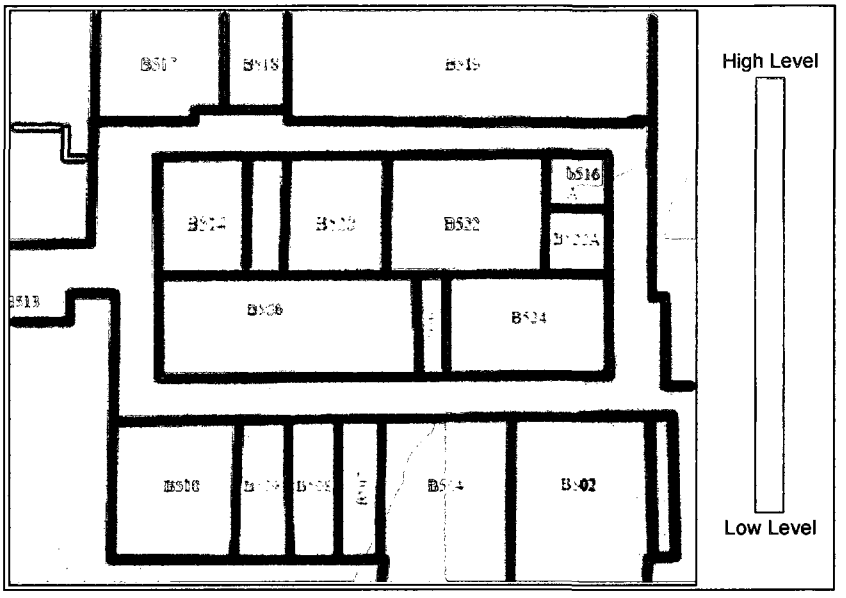



**Figure 5.22. Result of measuring RSS in the area. Note: there exists other Access Points on other floors of the same building, which has an effect on the result.**

Experiments were conducted in the three positions shown in Figure 5.19. The locations were chosen because the RSS varied between them. Position 1 has low RSS values; position 2 has medium RSS, and position 3 has high RSS. The tests were conducted on the system that included the RFID technology and the Conflict Resolving Center and on the one that did not. Results (Figure 5.23 and Figure 5.24) show a major increase in the accuracy at position 1: average error distance on the system that included the RFID and Conflict Resolving was 3.5 m. On the system that did not have the RFID and Conflict Resolving, the average error distance was 4.4 m. The increase in accuracy was, therefore, 0.9 m. Position 2 results show an increase in accuracy of 0.6 m. Position 3 results had almost no effect or increase in accuracy.

According to the T-TEST conducted on the collected data (see Table 1 for details), we see that at position 1, the result is 3.94% which is within the 5% limit, meaning we can

assume that using our system does have an effect in increasing location detection accuracy in areas of low availability of access points and weakest RSS values. Conducting a T-TEST on collected data at position 2, the result is 8.46% which is slightly higher than the 5% limit, meaning that using our system may have an effect in increasing location detection accuracy in areas of medium availability of access points and medium RSS values. More data must be gathered to give a final conclusion in such areas. Finally, according to the T-TEST conducted on collected data at position 3, the result is 93.02% which is much higher than the 5% limit, meaning that using our system in areas of high availability of access points and strongest RSS values does not have an effect in increasing location detection accuracy.

The main observation is that there is a significant benefit in using our system in areas of low availability of access points and weakest RSS values. This advantage not only affects location tracking but also assists the Dempster-Shafer algorithm to increase the accuracy of location prediction.

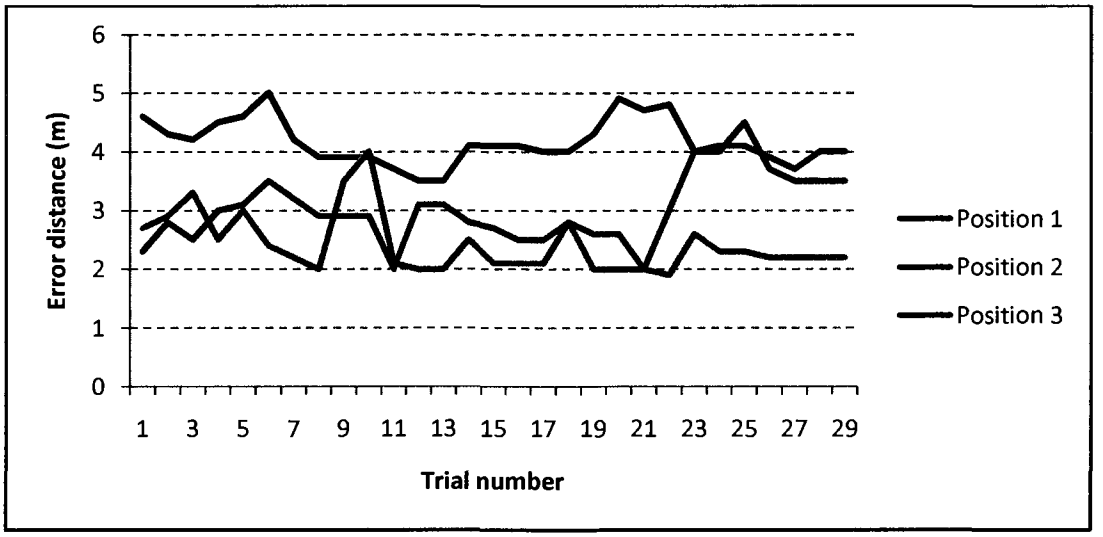



**Figure 5.23. System accuracy without RFID and Conflict Resolving.**

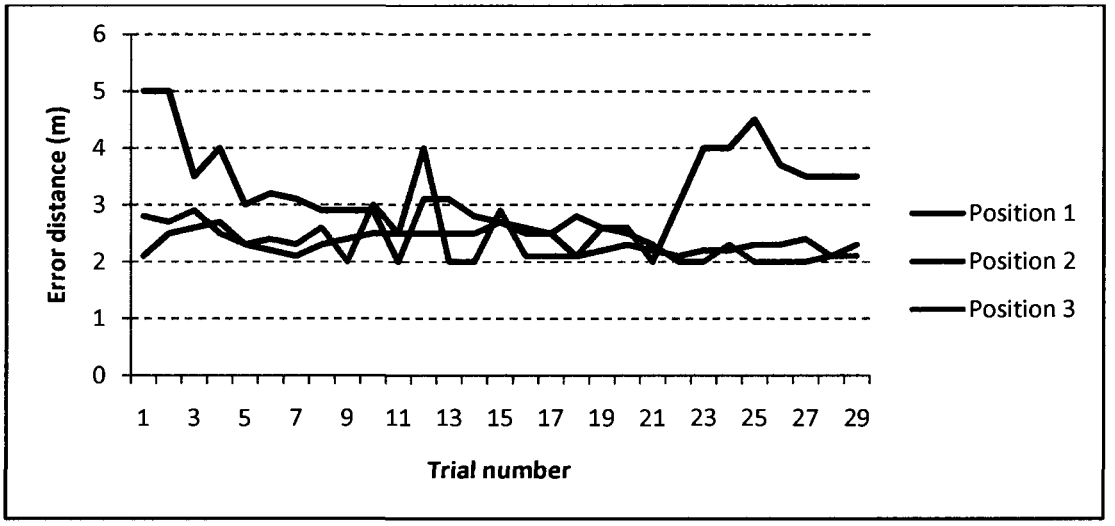



**Figure 5.24. System accuracy with RFID and Conflict Resolving.**

**Table 1. T-TEST conducted on collected data at the three positions.**

| | *Position 1A* | *Position 1B* | *Position 2A* | *Position 2B* | *Position 3A* | *Position 3B* |
|---|---|---|---|---|---|---|
| **Mean** | 4.43 | 3.53 | 3.25 | 2.61 | 2.706667 | 2.67 |
| **Variance** | 2.354586207 | 3.117344828 | 1.956379 | 2.034724 | 2.764092 | 2.66631 |
| **Observations** | 30 | 30 | 30 | 30 | 30 | 30 |
| **Pooled Variance** | 2.735965517 | | 1.995552 | | 2.715201 | |
| **Hypothesized Mean Difference** | 0 | | 0 | | 0 | |
| **Df** | 58 | | 58 | | 58 | |
| **t Stat** | 2.107331356 | | 1.754665 | | 0.086182 | |
| **P(T<=t) one-tail** | 0.019711536 | | 0.042299 | | 0.465809 | |
| **t Critical one-tail** | 1.671552763 | | 1.671553 | | 1.671553 | |
| **P(T<=t) two-tail** | 0.039423072 | | 0.084598 | | 0.931619 | |
| **t Critical two-tail** | 2.001717468 | | 2.001717 | | 2.001717 | |

According to the scenario conducted, we evaluated the delay of the system, including the video session handoff [7] once the user arrives to the office. The average delay of the location tracking algorithm is 3500 ms. Table 2 provides detailed results.

**Table 2. Delay of System.**

| Entity | Operation in scenario | Performance |
|---|---|---|
| Laptop ↔ AP | WLAN/RFID scan & association | 400 ~ 1000 ms |
| Laptop ↔ Server | Communication delay b/w laptop & server | 750 ~ 1000 ms |
| Location Tracking | Location Tracking and Decision | 3000 ~ 4000 ms |
| Laptop ~ Desktop | Video Session Handoff | 895 ~ 925 ms |

Simulations performed by our lab [8] on the prediction algorithm have shown that accuracy in predicting the future locations of users increases with the users' level of predictability. Users with dense schedules, such as undergraduate students, have a high level of predictability compared to users with a low level of predictability, such as graduate students. The strongest piece of evidence comes from users' schedule constraints (see equation 11). Simulation results show that predicted future locations for an undergraduate student has an accuracy of 93%; for a graduate student, the accuracy is 65%. On-going improvements to the location predicting algorithm are currently in process.

Preliminary tests were carried out to evaluate the effectiveness of information delivery. The effects of exchanging large profiles and views are minimized because exchanges are performed only once. Users are required to supply their identification only at the time of initial contact. The "Ontology Accessor" retrieves a user's profile from the ontology to "Initialization and Filtering," "Location Tracking and Prediction," and "Location Conflict Resolver" Centers. The result is a thin client, an advantage for cell phone and Smartphone users.

## 5.6 Summary

In this chapter, we presented a prototype system implementation representing our architecture. The majority of the components were implemented in Java 1.6. The mySQL-based temporary database was updated with all context information arriving from WIFI access points and RFID readers. Our Ontology was modeled using the OWL language. Our examples illustrated the use of the ontology. A graphical user interface was developed to make the system more user-friendly.

Evaluating the performance of the location-aware system is extremely difficult. Therefore, we have limited our evaluation test-bed to a single floor only. We have provided a prototype implementation that shows the feasibility of using a location-aware user tracking system.

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

The goal of this thesis was to develop a system that enables the user to interact with the surrounding environment based on his or her location. We found three main problems with the majority of current location-aware systems. The first problem was most systems ignore the use of ontologies to express inter-context relationships that exist in most types of these systems. The second problem was the lack of use of multiple technologies, such as RFID, WLAN, Infrared, in detecting user location. The third problem is that many location aware services require a system to take action before the user arrives at the destination. To provide such service pre-configuration, the location-aware system requires a location prediction algorithm to accommodate such services.

We have, thus, provided a generalized solution to these three issues. We integrated our system to be used with an ontology context model using the OWL-language. This ontology model solves many of the problems with currently existing non-ontology based solutions, which lack a clear method of modeling knowledge within location-aware systems and are not easily sharable and extensible. Our ontology played the role of a repository to store context related to locations, users, environment surrounding, and so on. To achieve the easy communication between detecting and predicting location, those entities within location-aware systems require a common understanding of how key

concepts and knowledge is represented. Therefore, we have based our model of acquired context on ontologies. Many existing ontologies were too restrictive for the domains their designers built them to meet, or they were too abstract and difficult to extend in order to be useful in diverse context-aware systems. Therefore, our ontology provides an easy solution to modeling context knowledge by providing an ontology that was neither restricted to specific domains nor too abstract for easy use and extension to other domains.

The problem of a lack in use of multiple technologies in detecting user location problem led us to the conclusion that there should be a system that used two different but related methods for estimating the location of a user. As a result, we developed a system that explores two specific technologies for location sensing, WLAN and RFID. The first approach consists of discovering the surrounding access points and measuring the signal strength beacons emitted by each access point to the wireless mobile device. The measured signal strength is then mapped onto a two-dimensional map. The second approach improves the location accuracy by using active RFID tags, where the signal strength of beacons emitted by a RFID reader are measured and mapped onto the two-dimensional map.

Once the current location of a user has been identified, the problem of next location prediction is solved by having a "prediction center" that identifies the next location of the user using the "Dempster-Shafer theory" algorithm. The algorithm has the ability to gather pieces of evidence to help choose a location between different potential future locations. The reason behind using a next location prediction method is to reinforce the system into taking actions before the user arrives at his/her destination. Those actions

involve both the network level (e.g. network handover management [9]) and the service level (e.g. ontology-based negotiations, seamless video handoff).

## 6.2 Future Work

There are several remaining issues that require closer investigation. Given the difficulty of the concepts introduced within our location prediction architecture, the prediction algorithm remains an area in need of greater research. We are planning to further explore this field in the near future, and on-going improvements to the location predicting algorithm are currently in process. Our future work further develops the project by integrating other applications in order to provide a wider variety of location-based services. We will also integrate our system with the IMS architecture [7] to provide more applications and services.

In summary, we are constantly reviewing our proposed system, adding and modifying its structure in search of an application that can meet the requirements of users in all location-aware systems, regardless of their domains or capabilities. We believe that the current system can sufficiently meet these basic requirements. We continue to improve on this system by adding new options and capabilities as our research progresses.

# Appendix A

## A.1 Review of the Dempster-Shafer Theory

The Dempster-Shafer Theory is a mathematical probability theory of evidence that combines pieces of evidence to reach decisions in situations of uncertainty. It has attracted considerable attention because it outperforms other mathematical theories like Bayesian [37], because in the Dempster-Shafer theory evidence can be associated with several possible events instead of only one event. The main advantage of the Shafer theory over other approaches is its ability to model the narrowing of a hypothesis with the accumulation of evidence and to represent uncertainty in the form of reservation of judgment.

The Dempster-Shafer Theory is based on four concepts: the Basic Probability Assignment function (BPA or m), Belief Function (Bel), the Plausibility Function (Pl), and Evidence Combination.

The Dempster-Shafer theory of evidential reasoning starts by assuming a Universe of Discourse $\Theta$, which is a set of mutually exclusive propositions about a domain. Let $2^{\Theta}$ denote the power set of $\Theta$. The BPA in equation (1) defines a mapping of the power set of X P(X) of the interval between 0 and 1. The basic probability assignment is also referred to as the *mass distribution* (m) to distinguish it from the probability distribution. The basic probability assignment must satisfy the empty set condition in equation (2) and the condition where the sum of all the subsets of X is 1 in equation (3).

$$m = P(X) \rightarrow [0,1] \tag{1}$$

$$m(\emptyset) = 0 \tag{2}$$

$$\sum_{A \in P(X)} m(A) = 1, \tag{3}$$

where 'A' is an element in the power set. The *Belief Function* in equation (4) calculates the minimum degree of belief for a hypothesis A, while *plausibility* in equation (5) calculates the maximum degree of belief for that hypothesis.

$$Bel\ (A) = \sum_{B|B \subseteq A} m(B) \tag{4}$$

$$Pl(A) = \sum_{B|B \cap A \neq \emptyset} m(B) \tag{5}$$

## A.2 Evidence Combination

The Dempster rule of combination provides a means of combining degrees of belief for a related hypothesis from distinct sources of evidence. It does so by emphasizing the agreement between multiple sources and ignoring all conflicting evidence through a normalization factor K in equation (7). Suppose *mi* and *mj* are two BPAs of the same universe set Θ but from independent bodies of evidence, *i* and *j*. The combined BPA can be expressed as:

$$mi \oplus mj\ (C) = \frac{\sum_{X \cap Y = C \neq 0} mi(X) mj(Y)}{1 - K} \tag{6}$$

where,

$$K = \sum_{X \cap Y = \emptyset} mi(X) mj(Y) \tag{7}$$

where X and Y are all the possible subsets from the power set P(X), and the denominator 1- $K$ is a normalization factor making the sum of $mi \oplus mj\ (C)$ range between 0 and 1.

The theory's combination rule computes a measure of agreement between two bodies of evidence concerning different propositions discerned from a common frame of discernment.

Once evidence is accumulated using the combination rule, choosing which hypothesis is the most appropriate is decided using the calculated belief values with equation (8):

$$D = arg_{A \epsilon \theta} \left( \max \left( Bel(A) \right) \right) \tag{8}$$

where 'arg' represents argument.

### A.2.1 Applying the Dempster-Shafer Theory

The prediction center of the system architecture has access to the information from the ontology/repository that is needed to predict a mobile user's future location. The user's profile repository includes the context required for the system to collect evidence pointing to the future location. This evidence includes the user's schedule, tasks, interests, activities, current location, and location history. With this information, the

Evidence Extraction Component generates hypotheses and bodies of evidence that are used by the Mobility Prediction Class to calculate the future location.

#### *A.2.1.1 Information Gathering*

Before conducting any type of prediction, capturing the necessary contextual information is an important step. Environment and user context is needed to collect evidence pertinent to the user and the environment in the surrounding; in our case, context is stored in our ontology. Environment context describes information regarding the space a user occupies and surrounds. Information described in Section 3.2.1 concerning location ontology describes the environment context, while user context is described by anything that is related to the user, such as schedules, activities, interests, tasks, and much more.

#### *A.2.1.2 Evidence Extraction*

Now, to apply concepts of the Dempster-Shafer theory to information stored and gathered in the ontology, Evidence Extraction is applied to generate hypotheses and bodies of evidence that will form the input to the Mobility Prediction Class in the final step. Evidence Extraction is applied on the four types of premises described next, for each of which a frame of discernment $\Theta$ is generated from all potential future destinations:

$$\boldsymbol{\Theta} = \{O_i : O_i \in O\}, \tag{9}$$

where $O$ represents the target locations of the user.

The four premises that are used by the Mobility Prediction Class to construct the frame of discernment into groups of candidate future locations are:

- **User's Interests**:

A set of all user's interests, such as having coffee breaks, attending conferences, exercising, is constructed in the form: $I = \{I_1, I_2, \ldots, I_n\}$. Each interest $I_i$ is given the following characteristics:

- i represents a user's interest.
- $t^{(i)}_{earliest}$ , $t^{(i)}_{latest}$ , which represents the earliest and latest preferable time of the day to perform this activity.
- $t^{(i)}_{duration}$ represents the average time required by the user to perform an activity.
- $V^{(i)}$ represents the degree of importance of the activity for the user.
- $C^{(i)} = \left\{c_1^{(i)}, c_2^{(i)}, \ldots, c_n^{(i)}\right\}$ the set of characteristics related to an interest.

A set of evidences based on the user's interests is defined as $E^I = \{E_1, E_2, \ldots, E_k\}$ such that:

$$E^I = \left\{E_i : \forall I_i \in I \mid t^{(i)}_{earliest} < t_{prediction} < t^{(i)}_{latest} \text{ V } t^{(i)}_{duration} < t_{AvailableTime}\right\} \quad (10)$$

where $t_{prediction}$ is the current time of the prediction process, and $t_{AvailableTime}$ is the user's amount of spare time before their next scheduled meeting.

For each evidence based on an interest, a group of hypotheses $H^{(i)} = \left\{H_1^{(i)}, H_2^{(i)}, \ldots, H_n^{(i)}\right\}$ is constructed such that:

$$H_j^{(i)} = \{O_K : O_K \in \Theta \mid c_j^i \in C_o(O_K)\},$$

where $C_O$ represents a function of a set of characteristics. A belief mass value $m$ is associated with $H_j^{(i)}$ such that:

$$m\left(H_j^{(i)}\right) = \frac{1}{n} \tag{11}$$

- **User's Schedule Constraints**:

User's schedule constraints represent more evidence that can help in predicting future locations. For each piece of evidence based on a user's schedule constraint, a group of hypotheses $H^{(i)} = \left\{H_1^{(i)}, H_2^{(i)}, \ldots, H_n^{(i)}\right\}$ are constructed such that:

$$H_j^{(S)} = \left\{O_K : O_K \in \Theta \mid \left(t(O_C, O_K) + t(O_K, O_S) \leq j \times \frac{t_s}{n}\right)\right\}, \tag{12}$$

where $O_c$ is the user's current location, and $t(O_C, O_K)$ is the time that it takes the user to go from $O_C$ to $O_K$. $O_s$ represents the location of the earliest scheduled appointment, and $t_s$ the time left before the scheduled appointment takes place. A belief mass value $m$ is associated with $H_j^{(s)}$ such that:

$$m\left(H_j^{(s)}\right) = \frac{1}{n} \tag{13}$$

- **User's Goals and Tasks.**

A set of all user's goals and tasks are constructed in the form: $G = \{G_1, G, \ldots, G_n\}$. Each goal, $G_i$, is given the following characteristics:

- $t^{(i)}_{earliest}$ , $t^{(i)}_{deadline}$ , represent the earliest and latest preferable time for the goal to be accompished.
- $t^{(i)}_{duration}$ represents the average time required by the user to perform a task.
- $V^{(i)}$ represents the degree of importance of the task.
- $C^{(i)} = \left\{c^{(i)}_1, c^{(i)}_2, \ldots, c^{(i)}_n\right\}$ is the set of characteristics related to a task.

A set of evidences based on the user's goals is defined as $E^G = \{E_1, E_2, \ldots, E_k\}$ such that:

$$E^G = \left\{E_i : \forall I_i \in G \mid t^{(i)}_{earliest} < t_{prediction} < t^{(i)}_{latest} \vee t^{(i)}_{duration} < t_{AvailableTime}\right\}$$

where $t_{prediction}$ is the current time of the prediction process, and $t_{AvailableTime}$ is the user's amount of spare time before their next scheduled meeting.

For each evidence based on an interest, a group of hypotheses $H^{(i)} = \left\{H^{(i)}_1, H^{(i)}_2, \ldots, H^{(i)}_n\right\}$ is constructed such that:

$$H^{(i)}_j = \left\{O_K : O_K \in \Theta \mid c^i_j \in C_o(O_K)\right\} \tag{14}$$

A belief mass value $m$ is associated with $H^{(i)}_j$ such that:

$$m\left(H^{(i)}_j\right) = \frac{1}{n} \tag{15}$$

It is important to note that classes of context other than user's interests, goals, and schedules can be used to build other bodies of evidence pertinent to the user's predicted location.

The final step is to combine each pair of hypothesis-belief mass using equation (6) of the Dempster-Shafer rule of combination to produce a list of candidate future locations. A belief value associated with each candidate location is produced, describing the degree of support for each. The location with the highest belief value is chosen as the user's predicted future location.

# References


[1] E. Aarts, "Ambient intelligence: a multimedia perspective," *Multimedia, IEEE*, vol.11, no.1, pp. 12-19, Jan.-March 2004.

[2] M. Weiser, "Hot topics-ubiquitous computing," *Computer*, vol.26, no.10, pp.71-72, Oct 1993.

[3] "Merriam-Webster Online Dictionary, 2009," [online] Available at: http://www.merriam-webster.com/dictionary/location

[4] A. Dey, "Understanding and Using Context", *Personal and Ubiquitous Computing*, vol. 5, pp. 4-7, February 2001.

[5] M. Smith, C. Welty, and D. McGuinness, "OWL Web Ontology Language Guide. W3C Recommendation.", W3C Recommendation, February 2004, http://www.w3.org/TR/2004/REC-owl-guide-20040210/

[6] Y. Al Ridhawi, A. Karmouch, "Ontology-Based Context-Level Agreements and Negotiation Protocol," in *5th International Workshop on Next Generation Networking Middleware*, 2008.

[7] M. Rawashdeh, A. Karmouch, "Seamless Video Handoff in Session Mobility over the IMS Network," *IEEE Conference on wireless, mobile, and multimedia networks*, June 2009.

[8] N. Samaan, A. Karmouch "A Mobility Prediction Architecture Based on Contextual Knowledge and Spatial Conceptual Maps," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, Nov./Dec. 2005.

[9] N. Montavont, T. Noel, "Handover Management for Mobile Nodes in IPv6 Networks," *IEEE Communications Magazine*, Aug. 2002.

[10] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE Infocom 2000*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 775-784.

[11] A. Ferscha, W. Beer, W. Narzt, "Location Awareness in Community Wireless LANs," Technical Report, COMPAQ Computer Austria GMbH.

[12] S. Abdel-Naby, P. Giorgini, "Locating Agents in RFID Architecture," Technical Report DIT-06-095, *Informatica e Telecomunicazioni*, University of Trento.

[13] W. Griswold, P. Shanahan, S. Brown, R. Boyer, M. Ratto, R. Shapiro. T. Truong, "ActiveCampus: experiments in community-oriented ubiquitous computing," *Computer* , vol.37, no.10, pp. 73-81, Oct. 2004.

[14] M. Ratto, R. Shapiro, T. Truong, W. Griswold, "The ActiveClass Project: experiments in ecouraging classroom participation," *Computer Support for Collaborative Learning,* 2003.

[15] J. Moons, C. Backer, H. Mannaert, "Components for a pervasice information dissemination architecture," *IEEE Spectrum*, 2006.

[16] J. Orwant, "Doppelganger Goes to School: Machine Learning for User Modeling," master's thesis, Dept. of Media Art and Sciences, Massachusetts Inst. of Technology, Sept. 1993.

[17] D. Ashbrook and T. Starner, "Learning Significant Locations and Predicting User Movement with GPS," Proc. Sixth Int'l Symp. Wearable Computers (ISWC 2002), pp. 101-108, Oct. 2002.

[18] G. Shafer, A mathematical theory of evidence. Princeton Univ. Press, 1975.

[19] J. Chan, S. Zhou, and A. Seneviratne, "A QOS Adaptive Mobility Prediction Scheme for Wireless Networks," Proc. IEEE Global Telecomm. Conf. (GLOBECOM '98), vol. 3, pp. 1414-1419, 1998.

[20] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", *W3C Recommendation*, Jan 15, 2004. (URL: http://www.w3.org/TR/CCPP-struct-vocab/)

[21] K. Henricksen, J. Indulska, and A. Rakotonirainy, "Generating Context Management Infrastructure from High-Level Context Models", in *Industrial Track Proceedings of the 4th International Conference on Mobile Data Management*, 2003, pp. 1-6.

[22] A. Held, S. Buchholz, and A. Schill, "Modeling of Context Information for Pervasive Computing Applications", in *Proceedings of 6th World Multiconference on Systematics, Cybernetics and Informatics*, 2002.

[23] G. Klyne, "A Syntax for Describing Media Feature Sets", *RFC 2533*, Mar 1999. (URL: http://www.faqs.org/rfcs/rfc2533.html).

[24] M. Strimpakou, I. Roussaki, M. Anagnostou, "A context ontology for pervasive service provision," *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference*, vol.2, April 2006.

[25] X. Wang, D. Zhang, T. Gu, H. Pung, "Ontology based context modeling and reasoning using OWL," *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on* , pp. 18-22, March 2004.

[26] T. Gu, H. Pung, D. Zhang, "A Middleware for Building Context-Aware Mobile Services", in *Proceedings of IEEE Vehicular Technology Conference*, 2004, pp. 2656-2660.

[27] J. H. Gennari, et al., "The evolution of Protégé: an environment for knowledge-based systems development", *Int. Journal of Human-Computer Studies*, 58(1), Jan. 2003.

[28] K. Markus, "A Summary of the International Standard Date and Time Notation", University of Cambridge, Computer Laboratory, December 1995, (URL: http://www.cl.cam.ac.uk/~mgk25/iso-time.html).

[29] B. Rao, L. Minakakis, "Evolution of mobile location-based services," *Commun. ACM* 46, 12 (Dec. 2003), 61-65.

[30] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, "The Anatomy of a Context-Aware Application," *Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp.59-68.

[31] N. Marmasse, C. Schmandt, "Location-aware information delivery with commotion," *Proc. of HUC 2000 Springer-Verlag*, 2000.

[32] I. Getting, "The Global Positioning System," *IEEE Spectrum 30*, December 1993.

[33] B. Hoffman-Wellenhof, H. Lichtenegger, J. Collins, "Global Positioning System: *Theory and Practice," Fourth Edition.* Springer-Verlag, 1997.

[34] Network Stumbler: (http://www.netstumbler.com).

[35] R. Hartley, P. Sturm, "Triangulation," Computer vision and image understanding ISSN 1077-3142, INIST-CNRS, Cote INIST, 1995.

[36] RFcode RFID: (http://www.rfcode.com)

[37] E. Karni, "Foundations of Bayesian Theory," Johns Hopkins University, Aug. 2005.

[38] Protégé Ontology Editor: (http://protege.stanford.edu/).

[39] I. Al Ridhawi, M. Aloqaily, A. Karmouch, N. Agoulmine "A Location-Aware User Tracking and Prediction System", in *Proc. IEEE-GIIS*, 2009.

[40] Li Ding; Lina Zhou; Finin, T.; Joshi, A., "How the Semantic Web is Being Used: An Analysis of FOAF Documents," *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on* System Sciences, Jan. 2005.