ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULITY OF INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

Modeling Pervasive Context-aware Museum Guide Service (PCMGS)

By: Bethelhem Teka Telila

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

IN COMPUTER SCIENCE

December 2009

Addis Ababa

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULITY OF INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

Modeling Pervasive Context-aware Museum Guide Service (PCMGS)

By: Bethelhem Teka Telila

ADVISOR: Dejene Ejigu (PhD)

APPROVED BY:

EXAMINING BOARD:

1. Dr. Dejene Ejigu, Advisor ______________________

2. ______________________ ______________________

3. ______________________ ______________________

Dedication

To: The Almighty God

Acknowledgment

My first and deepest thank goes to the Almighty God. Had not been God with me, my dreams and personality would not have been real. I am very glad to tell the great care of God for anyone who will access my paper.

I am also grateful to my Advisor Dr. Dejene Ejigu, who lets my research idea become touchable. His close monitoring and supervision made this work what it is now. I would like to thank all instructors of the department of computer science, Addis Ababa University.

I am indebted to my mother Emiye; you are not only a mother, but also a friend whose intimacy is built in care and love. I never forget those days of my education that you passed through discomfort with my depression and happiness with my relaxation. Brothers and sisters, it is my deepest feelings to say you are my blessings from heaven. I am blessed to be a member of such a supportive family full of peace and fun.

It is a privilege to thank Solomon Abera, who boosts my courage to be a better person. You were really my engine throughout the course of this paper in supporting with moral and crucial materials needed at any point in time. Your love and support made me strong and confident. I am very lucky to have you by my side.

I am also thankful to Birhanu Mekuria, who shapes my mind by showing the challenges and successes while doing a thesis work. You were my mentor from sharing research ideas through providing reading materials to the way of having such a readable work. My classmates, especially those of you were around computer science thesis laboratory, I will never forget the time we had there. Thank you all! Finally I would like to thank my colleague Lemma Nigussie and my lifetime friend, Tesfanesh Kifle who were around me on different activities while working on this thesis work.

I love you all!

Table of Contents

List of Tables

List of Figures

List of Equations

List of Algorithms

Acronyms

AI: Artificial Intelligence

API: Application Programming Interface

CATIS: Context Aware Tourist Information System

CLDC: Connected Limited Device Configuration

CM: CoMment

CL: Confidence Level

FOAF: Friend Of A Friend

GPS: Global Positioning System

J2ME: Java 2 Micro Edition

J2SE: Java 2 Software Edition

LAN: Local Area Network

LOL@: Local Location Assistant

MIDP: Mobile Information Device Profile

NF: Negative Factor

PCMGS: Pervasive Context-aware Museum Guide Service

PDA: Personal Digital Assistant

PEACH: Personal Experience with Active Cultural Heritage

QoS: Quality of Service

RDF: Resource Description Framework

RF: Radio Frequency

RFID : Radio Frequency IDentification

TId: Tag Identification

UDDI: Universal Description, Discovery and Integration

UML: Unified Modeling Language

VId: Visitor Identification

VT: Visiting Time

Wi-Fi : Wireless Fidelity

Abstract

Enhancement of hardware technology with advancement in application of pervasive computing shapes mobile users to seek devices on their disposal which may accomplish their day-to-day activities. Pervasive environments are dedicated to serve their users through embedded and/or independent physical devices. These environments have knowledgebase to learn from. By incorporating the incoming user and environment context with the stored knowledgebase, they prepare services on behalf of their users. The number of users of such environments is different with respect to the area of application where the pervasive system gets deployed. If there is more than one user in the environment, interest of each as of their personal profile should be considered.

Among the promising application domains of pervasive computing, tourism is the one that is functional with multi-users (large number of tourists to be served at a time). Having little or no information about the area they are visiting, tourists are potential beneficiaries of pervasive guide systems. To this end, this paper reports our research effort that models a pervasive context-aware museum guide service along with its implementation detail. Our model discovers visitor preferences or prepares list of selected heritages from the museum customized as to visitor's profile, analyzes each preference or proposed heritage to assure if it is liked by the visitor or not, and then monitors associating guide service with the analyzed heritages. The research theme of our model grounds on visitor satisfaction and design of expressive museum environment which are directly related with quality of guide service.

Keywords: Pervasive Computing, Context-awareness, Museum Guide Service, Preference Discovery, Preference Analysis, Quality of Guide Service.

CHAPTER 1: INTRODUCTION

1.1 Background

Pervasive computing is a new computing trend that emerged out of distributed and mobile computing. Among the important characteristics of pervasive computing, context-awareness, invisibility, ad-hoc networking, and sensors are mostly mentioned. Context refers to information that is used to express a situation of an entity, where the entity can be a person, or any other thing. Context can be location, preference, emotion, or activity of the entity from which the context is captured. Hence, the ability of a given system to understand the context of an entity is termed as context-awareness. A typical pervasive environment comprises of tiny and embedded devices that tend to work well with no or minimal user interference through its invisibility nature. An ad-hoc network is characterized by the connection of devices without previously configured physical infrastructures, and an application or device detects context by sensors [4, 21, 31].

The “anywhere and anytime” foundation of pervasive computing makes its application domain immense. In addition this, the rapid growth in hardware technology and the number of users of mobile devices such as cell phones and PDAs (Personal Digital Assistant) allow pervasive systems to work fine while system components or devices are moving [12].

Office and home based systems [22], health care [23, 24], and tourism [12] are some of the promising and vast application domains of pervasive computing being explored. The principal context information used in context-aware tourism systems is the location of the tourist [25]. However, parallel to improving the level of context-awareness of a system, it is possible to incorporate personal interests of the visitor, his/her schedule, current time, language preference, and so forth. A context-aware tourism environment serves tourists. Context data extracted from the tourist profile along with other supporting knowledge sources such as information about sites to be visited, hotel services around and the like provide important input components of such environment.

Museum is a “permanent institution in the service of society and of its development, open to the public, which acquires, conserves, researches, communicates and exhibits the tangible and intangible heritage of humanity and its environment, for the purposes of education, study, and enjoyment” [32, 33].

Pervasive museum guide is specialization of pervasive context-aware tourism service where the pervasive environment is confined to a specific place and creates smart space. Smart space is a well-defined area, open or enclosed, that incorporates a collection of embedded systems (computers, sensors, user interfaces, and infrastructure of services) [31]. Hence smart museum is a confined pervasive environment where its visitors may get appropriate guide service adapted to their current context information. Among the different tasks accomplished in museum guide service, providing explanation about heritages registered in the museum is the foremost. For ease of exhibition, cultural, religious or other heritage types found in a museum are arranged in a certain pattern. These heritages narrate the living style of people in the country, such as their level of thinking with respect to culture, art, and religion [27]. Whenever the visitor joins the space where the pervasive context-aware museum guide service is deployed, the system shall lead a direction to start with, followed by context-oriented information displayed on his/her cell phone or PDA. Then the smart space tracks the location and other context of the visitor and explains the heritages placed on the shelf where the visitor stands. In this paper, we present a report on our research work which models pervasive context-aware museum guide service that proposes list of heritages to be visited by each visitor adapted to their current context, analyzes every proposed heritage if it really entertains the visitor to whom it is proposed, and renders guide service while controlling visiting session status of each active visitors found in the museum. The study incorporates wider range of visitor context data enabling to capture a detail knowledgebase as decision support element. The wider context property scope touched in our work aims at increasing the quality of guide service in terms of visitor satisfaction and building correct perception over heritages.

1.2 Motivation

Tourism is an area that benefits from systems known to be context-aware. Since the time of its invention, pervasive computing has been knocking the door of many application domains including tourism. Obviously tourism is functional with mobility of tourists having little or no information about the place or the object being visited. Hence, pervasive context –aware tour guide service helps tourists significantly. It is mostly implemented to provide tourist guide on the go [12, 13]. On top of this, the perception of visitors about the heritages being visited can be affected by the guide service provided. Therefore, the guide service prepared for visitors should meet their expectation to put a positive impact on the quality of guide service.

Furthermore, pervasive environments should act as to the interest of their users while assuring their spontaneous service is likely enjoyed by them or should request for user confirmation.

To this end, modeling context-aware museum guide service focusing on visitor satisfaction and constructing expressive museum environment forms the motivation of our work.

1.3 Statement of the Problem

The major service rendered in museums is a guide service (explaining heritages exhibited on shelves). The quality of the guide service determines visitors' perception on heritages and their satisfaction. A museum environment assisted by human guidance imposes many problems into the guide service by which most of them are due to human factors. Negative or unenthusiastic visitor guide practices, such as getting bored due to role strain and then ill-define historical cultural or religious background of a heritage, being too fast to be followed by the visitor or too slow to precisely explain a heritage, and other negative guide practices affect the visitor perception negatively.

On the other hand, quality of a given service depends upon the degree to which the service is rendered in favour of the user situation. For instance: the quality of a dining service provided in hotels is partially determined by the extent to which the service is customized as to the food preference, and economical situation of their customers. Likewise, keeping visitors' interest is a core service quality parameter in museums.

Although different research works are done around pervasive museum environments, still there are visible gaps which need further research efforts. We have listed some of the potential research gaps that affect quality of museum guide services as follows:

- Lack of pervasive context-aware tour guide services dedicated in assuring if their service is really liked by their users with minimal or zero user destruction.
- Lack of pervasive context-aware tour guide services incorporating finer visitor and environment context (a wider visitor and museum context scope) in preparing proposed service.
- Lack of pervasive context-aware tour guide services that let visitors express their deep appreciation on heritages by spotting comments and incorporate these comment values into consideration for future guide service preparation.
- Lack of pervasive context-aware tour guide services that update their learning scheme dynamically to consider contemporary issues during service preparation.
- Lack of interactive pervasive museum environments that extend guide services according to questions raised by visitors about a given heritage.
- Lack of guide services that promote standardized heritage explanation to minimize wrong or under-valued visitor perception.

To this end, the core theme of this research work is the realization of a pervasive context-aware museum guide service that handles the aforementioned problems in current pervasive museum environments.

1.4 Objectives

1.4.1 General Objective

The general objective of this thesis work is to model and implement a pervasive context-aware museum guide service.

1.4.2 Specific objectives

The specific objectives of the research work are stated as follows:

- Studying the existing architectures of pervasive systems related with tourism specifically visitor guide services in museums.
- Identifying research components that potentially address the research problem of our work.

- Proposing a model that best suite a pervasive museum guide environment and focus on quality of museum guide service and visitor satisfaction, and
- Preparing Implementation detail and developing a prototype that demonstrates the proposed model

1.5 Scope and Limitation

The pervasive context-aware museum guide service model we proposed has two design phases namely, Guide Service Preparation phase and Question Generation phase. The first phase deals with procedures that range from discovering proposed guide service map (preparation of list of heritages selected to be visited by the specific visitor for whom the phase is initiated) to analyzing each heritage on the list to assure if it is really liked by the visitor. The output of this phase is net service map delivered to the visitor with the remaining components of PCMGS to start the visiting session. The second phase comes after the visiting session gets started. It deals with generation of list of questions prepared and sent to visitor's PDA so that the visitor may select possible questions of his concern.

Firstly, PCMGS (Pervasive Context-aware Museum Guide Service) extract profile of newly coming visitor while first joining from his/her PDA, and pre-process this profile to serve as an input in process of proposing list of heritages for the visitor. Another input for this process is organized as a dynamically updated learning set comprises of previous visitor profile history along with the proposed and net service map. Then it analyzes each proposed heritage to assure if it is liked by the visitor, and again filter it to net heritage list. Once the customized list of heritage is prepared, it track X-Y coordinate location of visitors, and then pre-process this coordinate point into relative location (high-level location) such as room number and shelf number. Next to this, it monitors the guide service using the list of heritages prepared as service site map. Finally it accepts comment on each heritage visited and update decision support parameters related to the heritage being visited by using values extracted from the visitor and/or PCMGS environment context, and comment provided by the visitor about the heritage. Further more, updating learning set (visitors profile data source) for future and generation of list of question are among the major activities after visiting begins.

Out of various reasons contribute to limitations on course of this research work; time is a crucial scared resource we had. As a result, some of the proposed research components are not included in the prototype implementation. The second limitation that hinders settlement of our prototypical setup is lack of technical equipments such as RFID readers, RFID tags and PDAs. To simulate the functionalities of these hardware equipments, we had been forced to develop a program that imposes additional burden on top of our work.

1.6 Methodology

To achieve the aforementioned objectives of the research, different activities grouped in phase are done. The methodology we use comprises of three phases. The first phase focuses on reviewing of literatures so as to have a deep understanding of pervasive computing. Activities in this phase are mainly studying state of the art in pervasive computing, and compiling the review.

The second phase is studying the proposed system. This phase acts as a blue print of the third phase. The major activities accomplished are preparing and conducting interview for National Museum of Ethiopia to capture raw facts that constitute a real museum, conducting site visit to come up with touchable guide service information, identifying system and user requirement, modeling a proposed architecture, capturing museum constructs using ontology modeling approaches, and devising a learning tools

Integrating results of the previous two phases, the last and the third one is to build the prototype to show the validity of the proposed model. Studying tools, technologies, and protocols required to develop the prototype is the first activity. Then implementing the prototype follows and then finally demonstrating is done.

1.7 Application of the study

The result of this research can be used to take tourism domain one step ahead by:

- Improving the visitor guide service with minimal involvement of museum assistants.
- Improving the quality of the visitor guide service by providing visitor-specific guide service with minimal or zero user destruction.
- Reducing the risk of transferring wrong knowledge about heritages which affect the historical, cultural, and religious perception of the visitors for the country.

In addition, it can be adapted to city or country wide tour assisting system incorporating the context information of other entities such as hotels, main roads direction and traffic flow and so on.

1.8 Document Organization

This thesis document has six chapters excluding the current chapter. State of the art in pervasive computing is discussed in chapter 2. Pervasive computing, context, context-aware systems and their elements, context modeling and reasoning, and context-awareness in tourism domain are its major sections. Chapter 3 deals with common features of others' work related with this thesis. Chapter 4 discusses the detail of the proposed model, provides architecture holding components of proposed model, and scenarios to dictate overall accomplishment of the model. Demonstration of model prototype with general description of implemented components and their algorithm, tools and technologies manipulated, and demonstration is given in chapter 5. Chapter 6 provides conclusion of the study and future work. Finally reference and appendices close this thesis report.

CHAPTER 2: STATE OF THE ART IN PERVASIVE COMPUTING

In this chapter, concepts around pervasive computing are discussed to make readers more comfortable with its bolts-and-nuts which let them understand the proposed model to be dealt in the subsequent chapters. The review has flow of discussions from general concepts to specific details. Pervasive computing, context, context-aware systems and their elements, context-modeling, context-knowledge reasoning, quality of pervasive environments and context-aware tourism applications are dealt with one by one. Finally summary of the chapter is given.

2.1 Pervasive computing

The demand of remote communication across computer networks, fault tolerance, availability, and remote information access pushed centralized computing paradigm in to another computing culture, distributed computing. Mobility of users while accessing distributed services accompanied by mobile networks, mobile information access, and adaptive application comes up with vast research questions on mobile computing. Then after, pervasive computing comes as a third stage where context-awareness, ad-hoc networks, and smart sensor networks are performing to serve people on their surrounding environment. As pervasive computing is usually characterised by invisible components that can be embedded around different environments, it is sometimes referred to as "Anywhere Anytime Computing" or "Omni-present Computing" or "Ubiquitous Computing".

Desktop computing model is physical computing trend attached with conscious interaction of users assisted by graphical user interface to explore computerized services and resources. Pervasive computing is a post-desktop model in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone "entertaining" pervasive computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is in advancement from the desktop paradigm in that users have access to information and communications throughout their environment with minimal or zero interaction to the surrounding devices. This trend is particularly associated with the growth of wireless technologies that allow users to access online information and services remotely and synchronized data between different computers [29].

At the early days of pervasive computing, around 1991, integrating computing trends with the everyday lives of people was just fantasy. The vision articulated by Mark Weiser in 1991 claimed to come up with seamless computing environments that can serve people in their day to day activities. Pervasive computing era is categorized to third wave of computing technology since the time computer had been heard of. It is characterized by many computers embedded in the environment to serve one person. This era is preceded by mainframe computing era where many people shared one computer, through workstations and personal computing era in which one computer was used by only one user with the need of conscious interaction between the computer and the user [1, 21].

After a decade passed, the unimplemented vision of Mark Weiser gets a way through the availability of critical pervasive environment infrastructures and highly improved hardware technologies that were exotic around 1991. Commercial products like handheld and wearable computers, wireless LANs (Local Area Networks) and devices capable of sensing controlling user situation are some indications for the feasibility of pervasive computing applications. Different research papers and projects together represent a wide effort to make pervasive computing a reality [4].
Computer systems that surround, pervade, and intelligently serve people in friendly and non-destructive manner holds the imagination of many. And these systems are not designed to just fit the tradition of desktop computing. Expressing this imagination with a phrase leads to buzz-phrases such as ubiquitous computing, pervasive computing, invisible computing, disappearing computing, proactive computing, autonomic computing, ambient intelligence, and sentient computing. The next paragraph describes precisely each of theses buzz-phrase.

Ubiquitous computing, a phrase used interchangeably with pervasive computing, is all about embedding large number of computer invisibly so that the user gets services of those devices. This is the phrase in which computers are in the environment rather than representing environment using computers. Pervasive computing is also referred as the vision on devices or computers pervading lives. Invisible computing is a technology of enabling the user to have some task done and more burdens is laid on the task not the computer system the user interacts with. In other words, it is making computer systems carryout useful functions while acting it is invisible to the user in a manner that the user puts almost no focus on the hardware or the software components it comprises of. Disappearing computer is initiative of manufacturing artefacts which are commonly seen and used in everyday lives with computing capability embedded with them. Moreover, the artefacts are able to work together to result in new behaviour. Designing computer system that reduces user interaction and acts on his/her behalf is proactive computing. It leaves the user free from focusing on low level tasks. Autonomic computing, though similar to proactive computing in relating ubiquitous computing and utilizing context information of the environment and the user, is different from proactive computing in its focus on kind of system behaviour it is designed to achieve. Autonomic computing is self-monitoring, self-healing and self-configuring computer system. Building of intelligent user interfaces to obtain higher user friendliness and efficient services for the user is ambient intelligence. Sentient computing refers to a computer system that makes use of sensors and resource status data to keep a model of the world in which users and applications are sharing [2].

The aforementioned computing phrases overlap at the ultimate technology they are intended to create, being intelligent and pervade the user to serve. To achieve their objective, each of them is context-aware as a result of context information captured from the system environment and the user.

2.2 Context, context-aware systems, and their elements

The history of context-awareness started in 1992 on Active Badge project to introduce location awareness of the system so as to forward a call to the closest phone (among phones around) to the user [5]. The term context-aware is first used in literature in 1994 to describe context as location, identities of nearby people, objects and changes to those objects [8]. From then onwards different researchers redefined the term context as user's location, environment, identity, time, some others defined the term as the user's emotional state, focus of attention, location, orientation, date and time, and so on. In 1997, other researchers provided a more generic definition of context as the aspect of the current situation [9]. Another more accurate definition of context is provided by Dey and Abowd in 2001 as any information that can be used to characterize the situation of entities (person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves [6, 8].

- Context has fundamental properties with which it is characterized [3]. Context exhibits range of temporal characteristics to be either static or dynamic. Static context is stable and mostly it is defined one like user preference related to his/her profile. Dynamic context is changing frequently and it is mostly sensed context (are not defined and stored rather sensed to result in different data value at a given time).

- Context is often imperfect in that a given context claimed to define a particular situation of user or environment may not completely describe that specific situation. Such imperfection may come from limitation on the underlying technology of sensors or/ and context data representation.

- Context may be represented using different alternative approaches. For instance, location can be alternatively represented using latitudinal and longitudinal coordinates, street numbers, building names, room number and so on.

- Context is mostly interrelated with other context. A particular context or set of contexts can be used to result in other contexts. Here the resulting context is referred to as indirect or inferred context and the cause contexts are termed as direct contexts (they may be either sensed or defined contexts). For example, the location of a person can be sensed to be in bathroom, sensed context, the heating switch is on, sensed context, the bathroom door is closed, again sensed context and the user profile is checked to retrieve shower time preference, defined context. Incorporating these direct contexts another context can indirectly be derived, the person is taking shower.

From the characteristics of context its classification is straight forward. Context classification is depicted in figure 2.1.

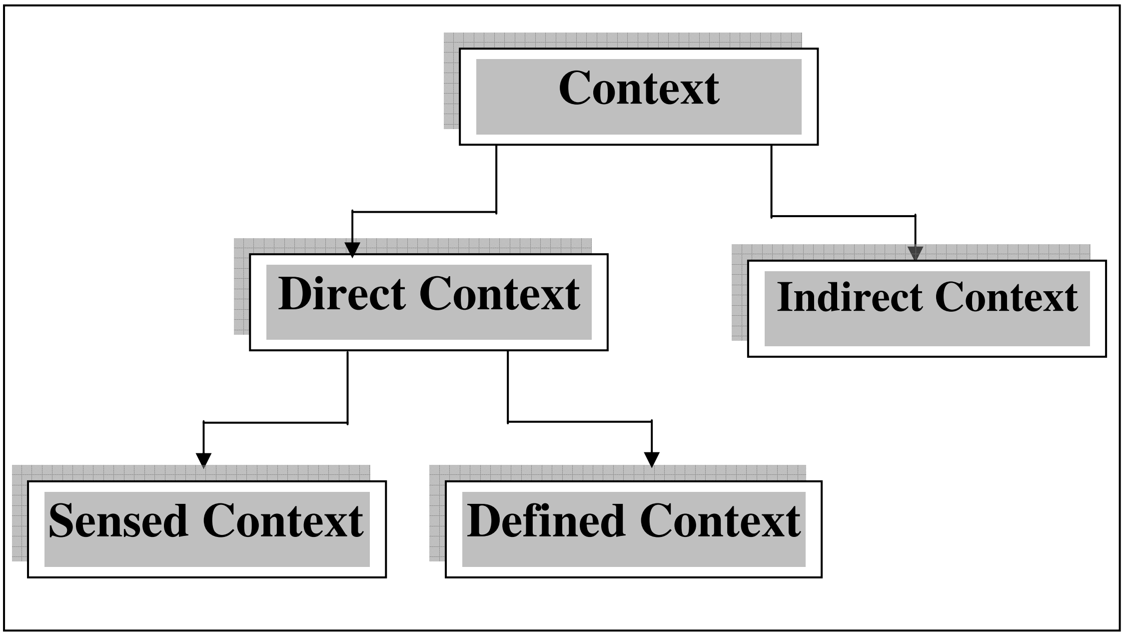


Fig 2. 1: Classification of Context

Context-aware system is a pervasive system that captures context information from system environments and users and act accordingly with the ever changing environment to result in the most favoured situation of the user. Context-awareness is the extent to which a pervasive system is user and environment aware. It can be immediately characterized by its context-scope. Scope of context refers to type and size of context properties supported by the system together with the ability to extend them to cope with unforeseen requirements [30]. Most context-data within a particular system is domain specific. Therefore, context-aware systems of different domain analyze different context information. Context-awareness of pervasive systems is growing rapidly in providing more amazing service for their users. Seng Loke describes the drastic improvement that can be achieved in context-aware systems, by saying "We are expecting a future full of aware systems, and once enough of them pervade our lives, we might actually become unaware of such aware systems. Mothers in the not-so-distant future can look forward to the aware bicycle and an enthusiastic cry from their child, "Look ma, no hands, and no hands needed!" [2, page 209].

Context-aware systems may have different system components while they are designed for different domains or different design goals with in the same domain. Some may have highly complicated sensing components for ease of reasoning and utilization; some other may have simplified sensing components with sophisticated pre-processing and reasoning components and so on. However, every context-aware system has sensing, thinking and acting elements and they share abstract and generic architecture of three layers while differing in system components at each layer.

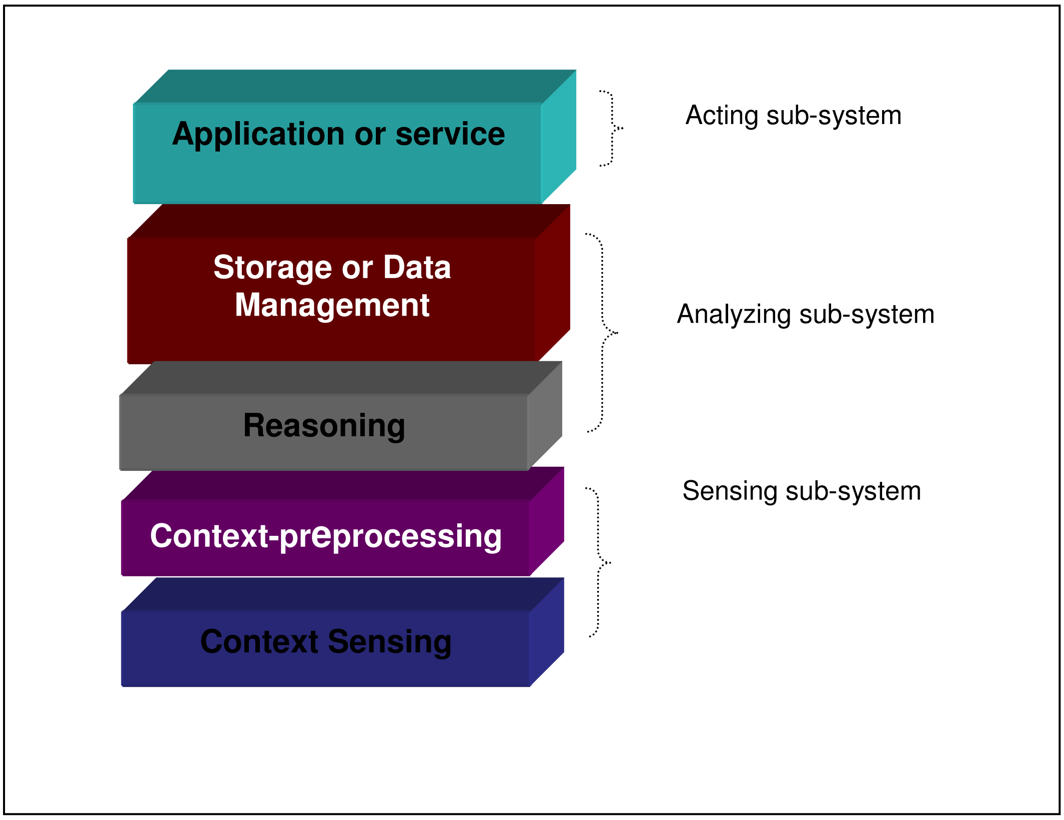


Fig 2. 2: Context-aware Systems Architecture Partially Adapted from [7]

Sensing sub-system: The lowest sub-system is for sensing and retrieving raw context data from physical or virtual sensors. Comprises of sensor networks designed to sense and retrieve raw context data. Physical sensors are major components of this sub-system designed to be very specific concerning the context data type they sense. According to their designed goal the preciseness of the collected data also varies. Context pre-processing component of this sub system is to rearrange raw context data into meaningful context information.

In tourism domain location of visitors is the most important context data tracked. Among the currently available physical sensors designed to sense location of objects, RFID reader is the one that works using Radio-Frequency signal. RFID readers function with RFID tags, which emit the identity information stamped on it to the requesting reader. Therefore the RFID readers together with RFID tags provide location of objects and their corresponding identity. The signal received by the readers is Radio Frequency (RF signal). Each reader (at least three), receives its own RF signal from the same object identity (stamped on the tag). Then controlling system collects the signals from the available readers and determines the location of the objects. Among different techniques used triangulation is one of them. Triangulation is a technique to determine location of unknown point by using two known baselines or coordinates. It deals with the measurement of the angle of the triangle formed between the observer and two known locations [36]. The readers send signal into tags frequently. The dynamicity (the frequency at which a context data is collected from the source) of the context gathering scheme can be configured as the device is first installed or while functioning. Network of such physical sensors raise the precision of context information and possibility of data communication.

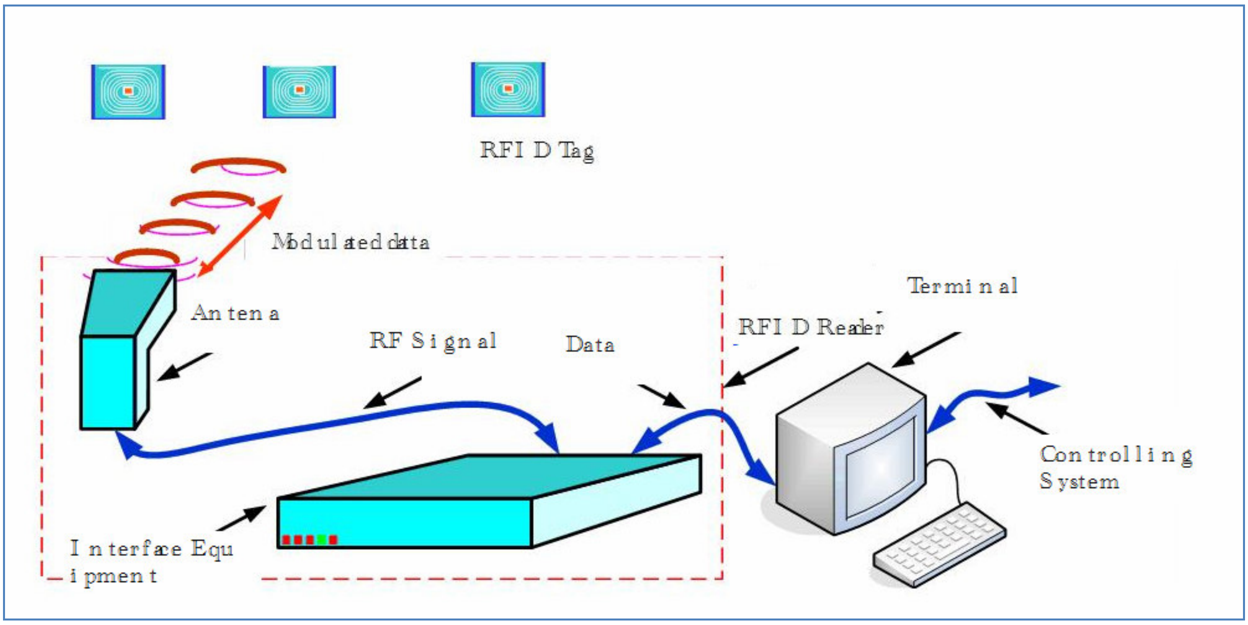


Fig 2. 3: RFID System Architecture [34]

Analyzing sub-system: The middle sub system comprises of reasoning and data management. The reasoning component is the core component from the architecture. It is done by techniques of different complexity ranging from simple event condition rule to sophisticated Artificial Intelligence (AI) logics. The central importance of reasoning part is to discover new knowledge out of the already existing learning set. The new knowledge is mostly used as crucial input in decision elements. The storage and data management component is responsible to handle context as well as other system data in a way they can be accessed and stored efficiently.

Acting sub system: makes use of processed context information and other data structures and algorithms that may be used as an input for different services of domain interest [10].

2.3 Context Modeling and Reasoning in Pervasive Systems

In pervasive context-aware system, the term context modeling refers to a formal specification of entities and their relationship. It is sort of data structure that abstracts the sensed entities' context. This abstraction, suitable for machine processing, should always be done before a context is stored or taken for some manipulation purpose.

Reasoning is a process of extracting new knowledge from the existing knowledge. When it comes to context reasoning, it is same process in which the knowledge is context. Usually, context reasoning in context-aware systems enables to derive high level context from low level context/s and reconciling context inconsistencies. The objective of context reasoning can be elaborated with this scenario. "*The motion detector sensor in the kitchen reports movements and the oven sensor reports that the oven is switched on, and then high-level context information on the user activity can be deduced: the user is cooking* " [11, page 1].

The later objective, reconciling inconsistencies among contexts is described in the following scenario. "*When sensors report that the user's PDA is located in the office while his mobile is at home, then it cannot surely be deduced where the user himself/herself is located. Further information may help to solve this inconsistency: When the key card of the user is used to open the office front door, then it can surely be asserted that the user himself is at the office (and even further, that he cannot use his mobile, so all incoming calls should be forwarded to the office telephone)"* [11, page 2]

There are different context modeling approaches with different features. Set -theory based model, markup-scheme model, object-oriented model, logic-based model, object-relational model, and ontology-based model. Of these modeling schemes, ontology-based model is the most suitable context representing approach in that it enables to define concepts without ambiguity, supports re-usability of already defined concepts and context-reasoning [11].

Related with context there are four dimensions to let research works have different capability and approach to answer their research question. These dimensions are ***context scope, context representation, context acquisition mechanism and context access method*** [30].

Context Scope: The scope of context refers to type and size of context properties supported by the system together with the ability to extend them to cope with unforeseen requirements. It is characterized by property, extensibility, chronology, validity and availability of context [30]. Each of these context properties can be associated with time chronology as another nested dimension. Related with the context scope, this paper provides multi-modal heritage explanation as to the citizenship element of user profile that lets visitors of different country origin may get different explanation on a particular heritage. Schedule of a visitor is handled to recommend which heritages are to be visited with the remaining time of the visitor. In case the visitor is too fast to finish the recommended heritages, he will get additional recommendation service if s/he is interested to stay there. The time chronology of the visiting speed is handled in relation with the schedule. Dynamic generation of question as per the visitor profile, and language preference of the visitor are also supported. There are also common context properties like location, current time, and back history of the visitor if s/he came before in which it can handle interruption at any point of the visiting session.

Context Representation: Symbolizing of the row context data prior to context management process. There are different context representations or modeling approaches varying in accompanying feature. Mark-up-scheme model, object-oriented model, logic-based model, object-relational model, and ontology-based model are some of them. Pervasive computing systems require context modeling approaches to fulfil different requirements [20]. Some of the requirements are outlined below:

Distributed composition: This requirement stems out of the fact that pervasive computing evolves out of distributed computing. It is the extent to which the context representation or model to work well with lack of central component responsible for the creation, deployment and maintenance of data and services. In ubiquitous computing composition and administration of a context model and its data varies with remarkably high dynamics in terms of time, network topology and source.

Partial validation: It measures the ability of the context model in validating context knowledge on structure and instance level against the context model in use. This requirement is particularly important because of the complexity of contextual interrelationships, which make any modeling intention error prone.

Richness and quality of information: The way the context model supports quality and richness of context information captured from sensors.

Incompleteness and ambiguity: The ability of the context model to cover the incompleteness and/or ambiguity of context information, for instance by interpolation of incomplete data on the instance level. Usually the set of contextual information available at any point in time characterizing relevant entities in ubiquitous computing environments is incomplete and/or ambiguous; especially the context sources are sensor networks.

Level of formality: The extent to which the context model describes the contextual facts and interrelationships in a precise and traceable manner. "For instance, to perform the task "print document on a printer near to me", it is required to have a precise definition of terms used in the task, for instance what "near" means to "me". It is highly desirable, that each participating party in a ubiquitous computing interaction shares the same interpretation of the data exchanged and the meaning "behind" it (so called *shared understanding*)" [20, page 47].

Applicability to existing environments: The level to which the context representation is applicable within the existing infrastructure of ubiquitous computing environments.

The feature to each of these pervasive computing requirements against different context data representation or modeling approaches is summarized in table 2.1.

Table 2 1: Context Modeling Approaches [20]

	Approaches				
Requirements	Mark-up Scheme	Graphical Model	OO Model	Logic Based	Ontology Based
Distributed Composition	+	-	++	++	++
Partial Validation	++	-	+	-	++
Quality of Information	-	+	+	-	+
Incompleteness/Ambiguity	-	-	+	-	+
Level of Formality	+	+	+	++	++
Applicability	++	+	+	-	+
Key: ++ compressive + partial – Limited or none.					

Among the different context modeling techniques, in our research work we propose an ontology for representation of context data management that facilitate context reusability, context derivation and reasoning on consistency of knowledge.

High level entities along with their interaction can be represented using Unified Modeling Language (UML) model. Class diagram represents object model, sequence diagram, activity diagram, and state chart represent dynamic behaviour of objects in systems. Ontology model can be mapped from object-oriented system models such as class diagram and/or sequence diagram [35]. Fig 2.4 depicts how to map from system model (Class diagram and Sequence diagram) to ontology model. As shown below, the messages "up" and "open" in the sequence diagram is mapped into Move and Open concepts of the ontology, cause relationship between the messages "up" and "open" into the association of type next.

"However, no events but Stop can be executed just after Move is executed because the domain ontology specifies that Move has only one outgoing *next* relationship to Stop. Thus the inference rule on the ontology suggests that there are no *next* relationships between Move and Open and some events should be added to keep semantic consistency of execution order *next* relationship. Obviously, in this case the engineer should add the message Stop between "up" and "open", which a Lift object sends to a Door object" [35, page 6].

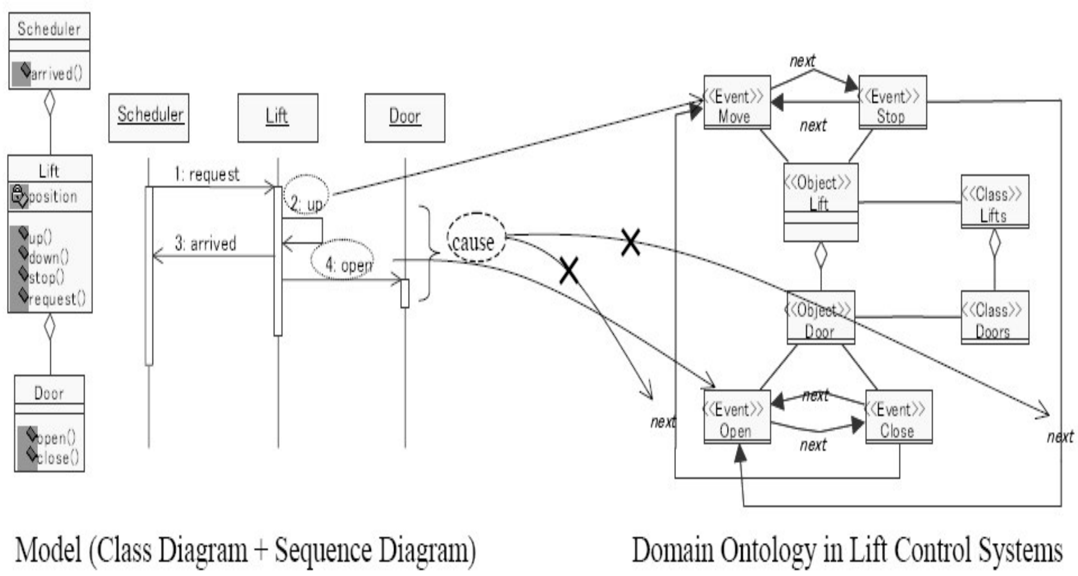


Fig 2. 4: Mapping between Object and Ontology Model [35]

Context Acquisition Mechanism: It refers to the automation of context acquisition and dynamicity in terms of when the context is acquired being either statically considered at system start up or dynamically at run-time. Full automation is a complete collection of context data out of manual control. Partial automation refers to the condition only partial context data is captured. And there is fully manual context data acquisition.

Context Access Mechanism: There are two generic ways concerning access mechanism, pull and push. Push access mechanism collects the context before it is required by the context services and pull access method collects the contexts information whenever it is needed there, having performance penalty in terms of response time [20, 32].

2.4 Quality of Pervasive Systems

Quality of Service (QoS) is a very subjective phrase to come up with quick definition. It is mostly used on internet and other networks to refer to the performance of the network measured in terms of parameters such as packet transmission rate, error rate, and other characteristics of achievement. It is also described as the ability to provide different priority to different applications, users, and/or data flows [28]. In broad terms, it is an abstract representation of the extent to which a given service stands to meet its user and application requirements. Therefore the issue of QoS can be dealt with at different level that a given service may comprise of. Most of pervasive context-aware services consist of three vast layers [7], context sensing, decision/reasoning layer and application or service level. Each layer has holes with respect to quality of service that requires research efforts. Context information refinement is an example of such efforts on context management layer that accordingly put positive influence on quality of service [3].

2.5 Context-awareness in tourism domain

The very nature of pervasive computing, "Being able to compute anywhere, anytime on any device" makes its general application domain to be "anywhere". Therefore there are extremely many areas of application where pervasive computing has already touched and many others to be explored in the future. Context-aware mobile applications tend to render services tailored to current situation of users while they are moving across different environment infrastructures. Tourism is an industry where tourists are enjoying one's property by moving from place to place. It is highly tied to information systems that provide guiding services and characterized by social interactions with friends, families and other tourists. Mobility of tourists together with the need to have social events demands adaptive service as to their current location.

There are research works to apply theoretical background of context-awareness in pervasive environment on tourism domain. Some of them related to this research work are discussed chapter three.

2.6 Summary

Pervasive computing is categorized to third wave of computing technology since the time computer had been ever heard of. It is characterized by many computers embedded in the environment to serve one person. Pervasive computing is also referred as the vision of devices or computers pervading lives. The term context-aware is first used in literature in 1994 to describe context as location, identities of nearby people, objects and changes to those objects. From then onwards different researchers redefined the term context as the aspect of the current situation. Another more accurate definition of context is "any information that can be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application, including the user and the application themselves." Context is characterized by its scope, representation technique, and acquisition method and access mechanism.

There are different context modeling approaches with different feature. Set -theory based model, markup-scheme model, object-oriented model, logic-based model, object-relational model, and ontology-based model of which ontology based model fulfills pervasive context-aware system requirements most [20].

Tourism is an industry where tourists are enjoying one's property by moving from place to place. It is highly tied to information systems that provide guiding services and characterized by social interactions with friends, families and other tourists. Mobility of tourists together with the need to have social events demands adaptive service as to their current position.

CHAPTER 3: RELATED WORKS

3.1 Overview

Once the idea of pervasive computing comes into play a number of research works have been conducted to make it adaptable for different application domains. Having comparable architectural layout, many of these works have different research components focused on context data storage, management, reasoning and usage. The difference in these components is an outcome of addressing generic problems across domains or adapting a generic architecture to specific domain of researcher's interest. On the other hand, being in the same domain, research works on pervasive computing may vary in principal issue they address. For instance, two distinct research works, being in the pervasive health domain, one may address issues of quality of context-information in terms of context data refinement and the other may address issue of decision support in terms of knowledge extraction. They may also differ in their software architecture. Besides, two research works, being in the same domain and addressing issue of decision support, may come up with two different architectural components to improve preciseness on decision. Types of context data captured, context modeling, and reasoning techniques are some of uniqueness sources to pervasive research works of the same domain.

Tourism is one of the application domains where pervasive research works dealt with to provide tourist with context-oriented service. Tourism services, especially museum guide, are of public interest. Hence many researchers dig out available pervasive computing technologies to be applied for this domain. In this chapter some of pervasive context-aware researches around tourism and related to our work are presented in two broad sections: Section 3.2 and 3.3. Tourist guide research works are grouped into the section 3.2. These systems are generally proposed for assisting tourists outdoor such as guiding in city map for roads, hotels, attraction centers and the likes. Section 3.3 deals with research works involving museum guide services in which the guide service scope is very specific to a confined environment.

3.2 Outdoor Tourist Information Systems

3.2.1 Cyber-guide

It is one of the earlier tour guide services. It is mobile context-aware tour guide with the focus of investigating context sensitive services. Originally developed for indoor tours and later extended to outdoors with GPS. The context data sensed is visitor's current location, as well as history of previous locations. Moving all of the paper-based information into a hand-held, position-aware unit provided a test bed for research questions on mobile, context-aware application development [16].

3.2.2 CATIS

A Context-Aware Tourist Information System, [12], is a web service based tourism information system. It is based on the scenario of a tourist is located in main road of Chicago metropolitan area, since it is noon the system requests restaurant information around the current location of the tourist, the retrieved information considers food preference and economical class of the tourist, finally, the resulting restaurant list is transformed into Web pages in a format appropriate for display of client wireless device. The elements of context in this work are location of a tourist, time of the day, speed and direction of travel, personal preferences, and device type. It describes how these elements are leveraged to adapt Web-based information that is delivered to mobile tourists. It has Web services-based system architecture having wireless client device for browsing, application server for coordination, UDDI (Universal Description, Discovery and Integration) server to act as services registry, context manager that manages both dynamic and static contexts, collection of web-service to provide different services.

Architecture of CATIS is indicated in fig 3.1. The links between components in the architecture numbered so that the sequence of information exchange will be maintained. The application server initially communicates with tourist's device to capture context data and hands to context manager. Then it integrates the output of context manager with the help of UDDI server and tourist information web server to serve the tourist via his/her device.

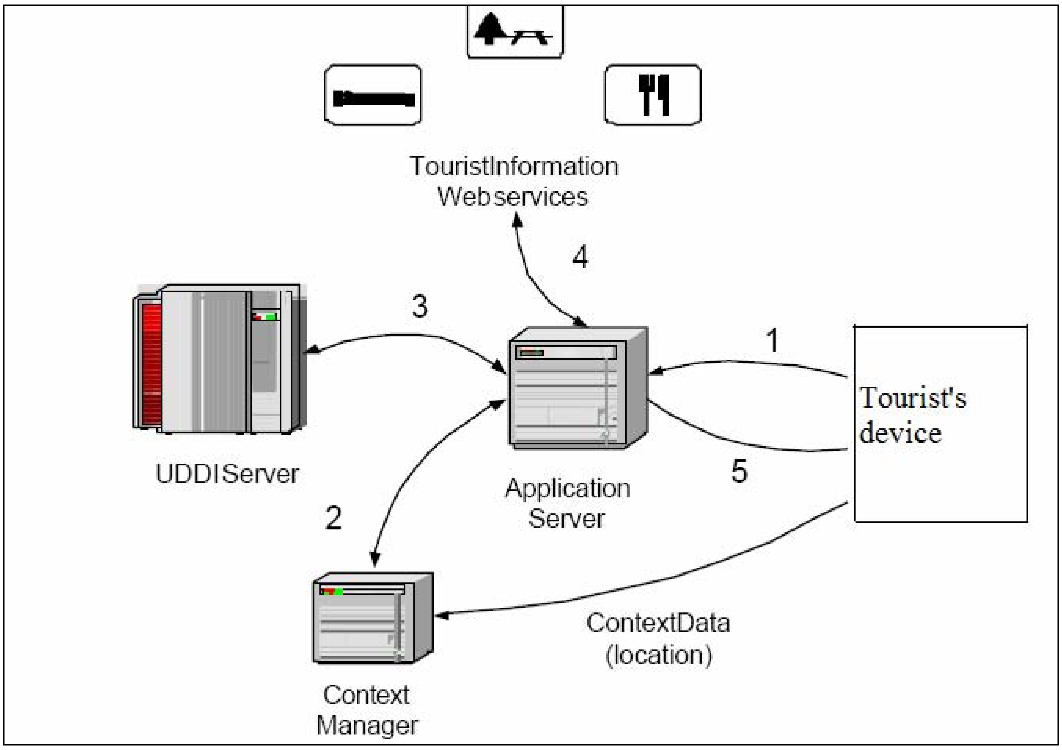


Fig 3. 1: Architecture of CATIS [12]

3.2.3 LoL@

It is an acronym for Local Location Assistant. It is a research work on a location-based mobile application, and providing tourists with multimedia tourism information about the city of Vienna. Its main focus is on supporting tourists during their trip with predefined tours, information about point of interests, routing functionality and multimodal interaction (e.g., speech control), based on a map.

3.2.4 Guide

It stems out of location-based services and its focus is to provide tourists with up-to-date and context-aware information about a city via their PDA, developed in the city of Lancaster, UK (United Kingdom), that provided an electronic handheld guide enabling visitors to Lancaster to access information about the city, to create tailored tours of the city, to access interactive services, to send and receive messages to/from their companions, to let other members of their group know their location and to leave virtual stickon notes at specific locations in the city so they can share their experiences with other tourists [17].

3.3 Indoor Tourist Information Systems

3.3.1 Semantic Web Technologies for Context-Aware Museum Tour Guide Applications

It is semantic web based and context-aware museum tour guide that adjusts its recommendations to the interests and preferences of individual visitors. Mainly it enables visitors to selectively share their experience with others [15]. It also deals with user specified privacy preference so as the visitors are confident about their information privacy for instance visitors can specify to make only their friends can see their exhibit rating report or access their current location. The authors claim that the service is built around an innovative semantic web framework that minimizes the development and maintenance costs associated with the introduction of new exhibits, new visitor-oriented services and new sources of contextual information. It has semantic web reasoning engine to infer high level contexts from low level contexts. Its system architecture is shown in fig 3.2..

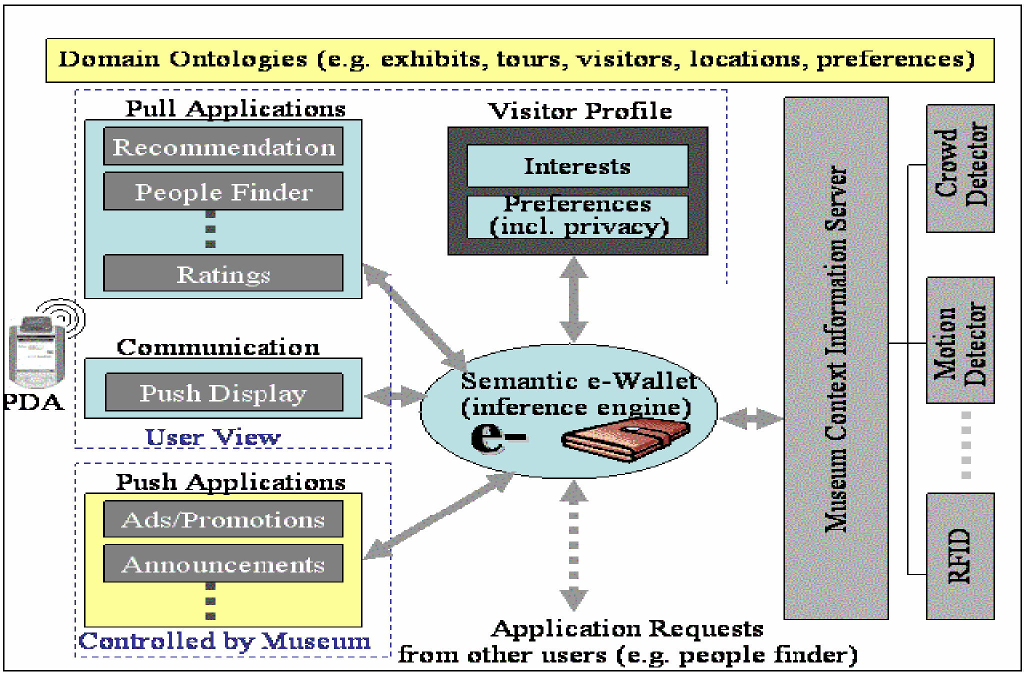


Fig 3. 2: Semantic Web Guide Architecture [15]

3.3.2 An Agent-Based Architecture for Museum Visitors' Guide Systems

Is agent based model focused on the design of “Active Museum" that interacts with its inhabitants [13]. It suggests a generic, layered architecture for agent based museum guide systems, and surveys architecture of some research works and projects on the area of visitor guide applications like PEACH (Personal Experience with Active Cultural Heritage) and Hecht museum system. Its generic architecture is supposed to facilitate research cooperation and also serve as a basis for the development of museum visitors' guide application. The proposed architecture considers interaction among the following generic agents

Spatial information mediator: responsible for reporting user current position periodically and/or on request. It provides positioning relations between visitors and museum exhibits.

User modellers that may include different agents implementing user modelling approaches, such as the collaborative approach, a stereotypic approach or an adaptable, personal user model.

Presentation composers may include agents that provide audio, text, slides or videos for presentation.

Information brokers that provide the information required for presentation composition: pictures, text and other media, stored locally or remotely.

Presentation clients - different clients with different presentation capabilities that may present information to users in the museum environment.

Information logger/analyzer is a set of development and research support tools.

3.3.3 A Context-Aware Smart Tourist Guide Application for an Old Palace

It is PDA-based location-aware tourist guide application for the old palace Deoksugung. It provides information like current location, attractions nearby, and detail about specific buildings. Rich multimedia support has been incorporated into the system to provide extra features to enhance the self-guide tour. Several modes of guide for different kinds of users approach are modelled. Provide a multi-lingual audio guide. The researchers of the paper claim about the uniqueness of their work, among number of research efforts into GPS-based hand held tourist application, in that the focus they have dealt with is on a location based practical application targeting the old palace Deoksugung in Seoul, and it also provides a multilingual audio guide service and multi-modal guide service depending on the kind of users as well [18].

3.3.4 Exploiting Social Context Information in Context-aware Mobile Tourism Guide

It is part of the joint project on context-aware recommender system and it has a principal goal of guiding tourists while considering the social interaction context [14]. Social interaction is defined as visiting interesting places with peers, sharing photos, buying souvenir for families, friends or spouses and so on. All these interaction is helped by the context-aware service. For instance, a tourist will be recommended interesting sites that are relatively nearby to his/her location, by considering the context of group members of the sites. Therefore the context element in this work focuses on the social context so that it is possible to make the tourist enjoy the current social interaction. It has ontology-based model for social context representing and reasoning, that make use of the popular FOAF (Friend Of A Friend) ontology that represents personal profile information and social interaction among groups of peers and built on Resource Description Framework (RDF) semantic web language. Hence the social context interpretation and exploitation process grounds from the FOAF graph representation that is augmented and enriched.

3.3.5 Supporting Small Groups in the Museum by Context-Aware Communication Services

Ahead of providing information on exhibits, it considers a small group that the visitor may be part of. They tried to facilitate interaction among members to stimulate further interest. It also provides an initial effort in allowing conversation among the visitors before and after the visiting session [19].

3.4 Summary

Research works discussed in previous section (CATIS, Cyberguide, and Guide) have one thing in common that is to help a tourist outdoor [12, 16 and 17]. The tourist joins their service on demand. They are not confined in a specific environment to give particular service rather track the services of other providers and adapt to the current situation of the tourist. They do not focus on specific system component to handle. Yet they just coordinate other's services to recommend in favour of the current context of their user.

The research works discussed in section 3.3, "Exploiting Social Context Information in Context-aware Mobile Tourism Guide", "Semantic Web Technologies for Context-Aware Museum Tour Guide Applications", "Supporting Small Groups in the Museum by Context-Aware Communication Services", [14, 15, and 19] focus on social aspect of museum guide service and primarily model social context data management over pervasive tourist guide environment. Social context data refers to the situation of a given user with respect to other users' context in order to have a particular service that concerns the situation of both parties. The context data captured in these works is coarse with respect to a given user. They have crude consideration of individual user context which hardly allow preparing adapted single visitor-based services.

"An Agent-Based Architecture for Museum Visitors' Guide Systems" proposes generic agent-based approach to design active museum guide [13]. It considers the common structure of museum guide services. It just proposes a skeleton architecture for agent-based museum guide services.

"A Context-Aware Smart Tourist Guide Application for an Old Palace", [18] comes very close to our work in that different modes of guide service for different visitors and a multi-lingual audio guide are supported. However it manipulates narrower scope of context properties. The wider the scope of the context property a service makes use of, the more it is adapted as to the favour of the user.

In this research work, multilingual museum guide service helped with huge knowledgebase of previous visitor history is modeled. Context scope-wise, the static context data considered includes language preference, heritage type preference, and visitor's country, visiting start time and schedule, back history of the visitor if he/she came to the museum before, already visited museums if any, educational background, and visitor nature (being optimist, pessimist or free). Dynamic contexts include location of the visitor with respect to shelves in the museum, visiting speed with respect to each heritage, and time chronology with respect to the visitor's schedule. Logical contexts that are derived from static and dynamic contexts are also important aspects of the guide service to be modeled. The knowledgebase is updated dynamically as each visitor finishes visiting. This is vital issue to adapt guide service preparation based on contemporary conditions of country or world that imposes direct influence on the museum situation. For instance: Assume a particular well-known heritage moved from its permanent location to the museum where our proposed model is deployed, and the number of its visitors increases that week. Learning this dynamically, our model takes shift of visitor interest into consideration and recommends the newly coming heritage for future visitors.

CHAPTER 4: PROPOSED MODEL

In this chapter, we present architectural components of PCMGS, and their detail description. The functionality of components is presented in two phases, namely service preparation and question generation phases. Service preparation phase comprises of tasks that discover visitors' preference or list of heritages to be visited by the visitor, analyze the proposed heritages and monitor the guide service. Question generation phase is handling questions while the guide service is continuing. Scenario is given at the end of each phase.

4.1 Architecture of PCMGS

In most pervasive context-aware systems, there are three major steps handled throughout service life time in the pervasive environment [3]. They are generalized as collecting context data, analysing it and providing service accordingly.

Increased life expectancy, mobility, value added feature, modularity, innate plasticity, interoperability, greater compatibility, and better flexibility are the common advantages of layered approach [37]. Hence, PCMGS is designed to follow layered approach comprises of three layers. The first layer, context sensing layer, is composed of sensors to produce raw context data that will be consumed by accompanying pre-processing components to result in logical context information. Context data management and decision are dealt in the second layer, context analysis layer. The third layer, service layer, is dedicated to provide service management. Fig 4.1 shows architectural components of PCMGS.

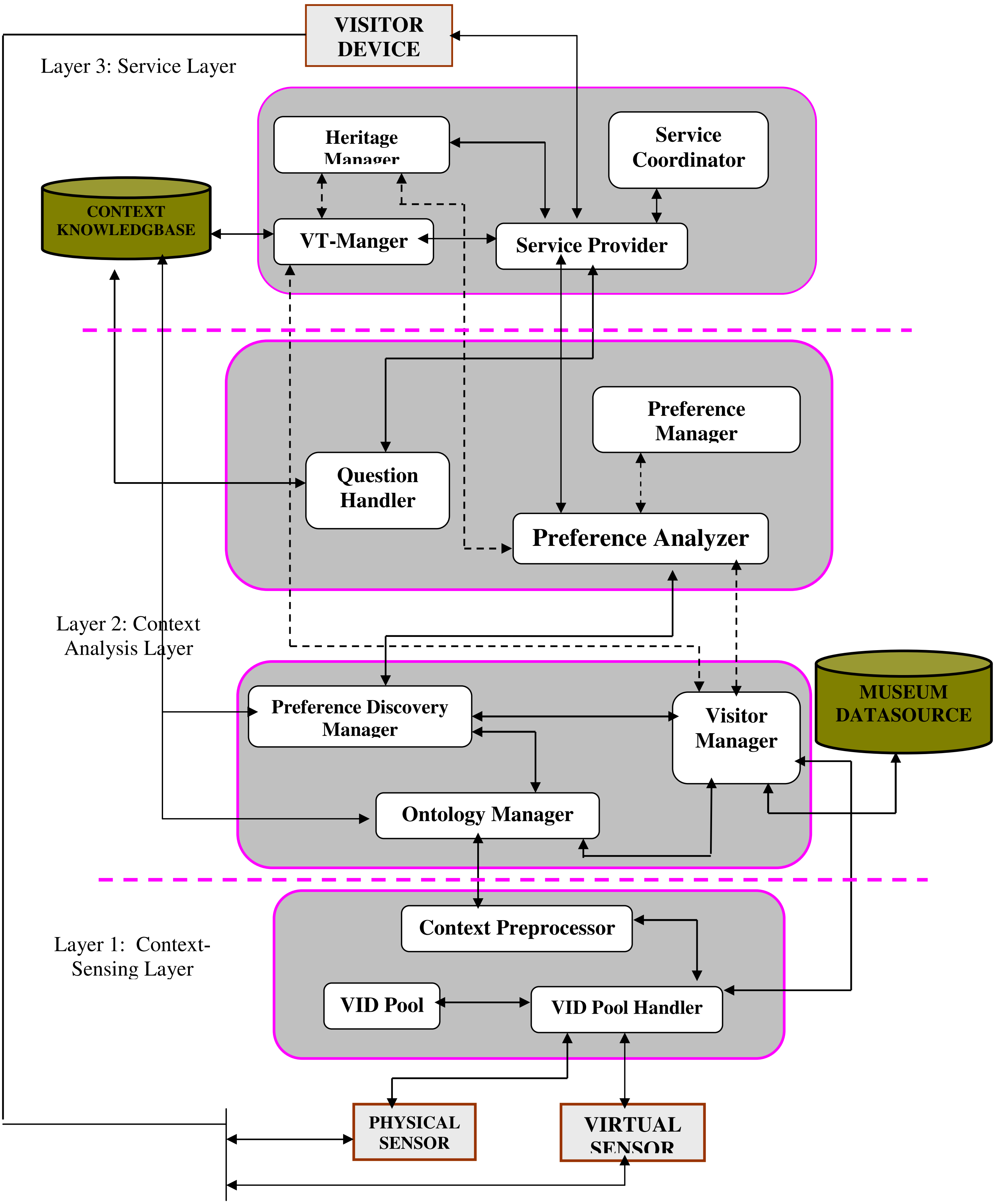


Fig 4. 1: Architecture of PCMGS

4.1.1 Components in Context Sensing Layer

- **Physical Sensor:** Being designed in variety of shapes and sizes, mostly these devices are designed to be small enough to get invisible in pervasive environments. Physical/hardware sensors are able to sense context data about the environment and its users. They are also designed for different purposes. For instance, location sensors sense user's X-Y-Z coordinates location point, temperature sensors measure the hotness or coldness of the pervasive environment or the objects around there, video cameras sense identity of objects interacting in the environment along with their movement pattern, and the likes. As per the requirement of the pervasive environment to be implemented, the type and number of hardware sensors may vary. The sensors needed in our research work are RFID (Radio Frequency IDentification) readers to track the location and identity of objects. Each visitor is given with RFID tag so that his/her location (in X-Y coordinate) will be traced by the RFID readers.

- **Virtual Sensor:** Usually referred to as software sensor and often retrieve static context data from different context sources such as user PDA. It is designed to run in parallel so that context data especially user profile from many visitors can be captured at a time. Software sensors can be deployed on application server or user device. On both cases its purpose is to extract context data from the source (can be hardware or software component). In our model software sensors will be deployed on PCMGS server and interacts to download visitor profile from their PDA. Unlike physical sensors, whose resulting context data is dynamic, software sensors produce static context data that will be stored permanently through persistent data management techniques.
- **VID Pool:** It is pool of TId (Tag Identification) mapped with respective RF signal (Radio Frequency signal) emitted from the physical sensors. The pool is constructed from list of (TId, location-point) pairs. Whenever new visitor joins PCMGS, s/he will be registered to the VID Pool. Physical sensors (RFID readers) continuously sense visitor's current location through the location of the RFID tag the visitor carries. Whenever the visitor finishes or interrupts visiting session, his TId (Visitor Id) is deleted from the VID Pool. Hence VID Pool contains only active visitors who are entertaining PCMGS. It is controlled and directly accessed by VID Pool Handler.

- **VID Pool Handler:** Manages VID Pool from appending (TId, location-point) pairs when individual visitors are first join PCMGS to deleting when they disconnected. It accepts continuous Radio Frequency signal and TId (Tag Identification) pairs from RFID readers. Then checks if TId exists in the VID Pool or not. If the TId does not exist, it appends the pair into the pool at the end (new visitor is being registered to the pool). If the TId already exists, it checks whether the respective location signal is changed or not. If the signal is not changed it waits till it receives another pair, otherwise updates the location-point value of the TId in the pool.

  Attached with the respective raw location context in terms of RF signals read from RFID readers, new TIds are submitted to Context-Pre-processor where as existing TIds attached with only dynamic context (location and visiting speed) data will be sent to Visitor Manager.

- **Context Pre-Processor:** It pre-processes raw context data to come up with logical contexts which are high level to hide un-necessary details. The major context to be processed here are location and visitor identification number. It collects inputs in the form of RF (Radio Frequency) signals for location context, processes the signals then outputs respective logical location. For instance: If this component accepts (TId, (RF1, RF2, RF3)) pair from the VID Pool Handler, where TId is the Tag Identification number and (RF1, RF2, RF3) are RF signal values from the readers with respect to TId, then it processes and determines the logical location of TId in terms of room shelf where the visitor is around. Once the raw location context is processed it is submitted to Ontology Manager. It also pre-processes (VId) Visitor Identification number from TId (Tag Identification number)

4.1.2 Components in Context Analysis Layer

Ontology Manager: Collects all contexts for the newly coming visitor and builds instances using the ontology based context knowledgebase. It maintains context data for these instances during the visiting session. It acts as a gateway across manipulation of guide services which are directly or indirectly dependent on the museum knowledge repository. This component therefore is responsible to query from the ontology repository (Context Knowledgebase).

Visitor Manager: It monitors active visitors during visiting session. It communicates with VID Pool Handler to destruct disconnected visitors from the pool and responsible to pass individual visitor information into different components such as VT-Manager, Preference Analyzer and Preference Status Manager. Here parallel context management of many visitors is exhibited.

Preference Discovery Manager: This component selects or discovered heritages to be visited by the visitor on his/her behalf. The selected heritages are all or usually part of the available heritages in the museum. The purpose of selecting heritages is to provide a heritage list that best suits the interest of the visitor so that the visitor enjoys more by having context oriented guide service discovery. It extracts list of user preferences with respect to heritages. The component learns from previous visitors' profile and corresponding guide service they offered. Visitors' profile consists of static visitor context data such as age, educational background, language preference, heritage type preference (archaeological, religious, artistic, and cultural and combination of any of these categories), already visited museum, visitor type or nature (being optimist, pessimist or free), schedule, and being new comer or not. The service they had offered is the heritage proposed by the PCMGS to the visitors. Any active visitor who is currently visiting in the museum will be added to the knowledgebase right after finishing visiting session. Therefore the discovery of service is based on a history of previous visitors that dynamically updated each time a visitor is served. This makes PCMGS to encompass up-to-date information to discover services to the newly coming visitor.

Preference Analyzer: The core research component of our model where each discovered heritage is decided to be implicit or explicit. A given discovered heritage is implicit if PCMGS has enough confidence level about its acceptance by the visitor for whom the heritage is discovered. This means, PCMGS is sure of acceptance of the heritage by the visitor. It is also confident in enjoyment of the visitor over that heritage. A given selected heritage is explicit if PCMGS has little confidence level about its acceptance by the visitor. Hereby, based on the implicitness of explicitness of a given selection, it adds the discovered preference automatically to the service cart or requests for approval respectively. The confidence level over acceptance of discovered services have threshold value that varies as to visitor type. Detail of this component will be discussed in section 4.5.1 (Service Preparation Phase).

Question Handler: It sends list of possible questions about the heritage being visited, into visitor's PDA. The list is prepared by PCMGS that takes visitor's context into consideration. Then the prepared list of question is sent to the visitor's PDA while visiting each heritage. Then it extracts respective answer by consulting Museum Data source and sends it to visitor's PDA. Detail of this component will be discussed in section 4.5.2 (Question Generation Phase).

Preference Status Manager: Preference is an abstract type that represents attributes and behaviours of visitor preference with respect to heritages. Its instances are frequently accessed by Preference Analyzer component to retrieve its important property, Confidence Level. Detail of this component is discussed in next chapter (Chapter 5)

Context Knowledgebase: The stored static context data, visitor profile, is central source of knowledge for important tasks accomplished in higher level along the architecture. Since the knowledgebase is dynamically updated whenever a new visitor finishes visiting, contemporary issues that affect the guide service will be taken into consideration. Consider the following scenario: *"John is red-handed while trying to steal fifty kilogram honoured golden cross from church. Then, after dealing with the church, the government announces to put the golden cross as a religious heritage in the museum where PCMGS is deployed. That week the number of visitors interested to visit the golden cross is increasing. PCMGS learns this current phenomenon from the profile of the coming visitors and uses this knowledge to incorporate in process of heritage preference proposal for the newly coming visitors."*

4.1.3 Components in Service Layer

Service Provider: It renders visitor guide service by mediating between visitors' PDA and remaining components of PCMGS. Basically, it sends heritage description in to visitor's PDA. Taking the service parameters handed from Service Coordinator, it consults Museum Data Source and serves all active visitors until the end of their visiting sessions. It is also an interface for comment and question.

Service Coordinator: Consults Service Provider after collecting important parameters to provide guide services to a given visitor. These service parameters are visitor identification number, list of heritages yet not visited out of the proposed and analyzed ones, current room number and shelf number. It treats only active visitor while visiting and disconnects them after finishing.

Heritage Manager: Represents heritages found in the pervasive museum environments. It comprises of important heritage abstracting parameters of which comment value is the most important one for our model. Detail of this component will be discussed in section 4.5.1 (Service Preparation Phase).

VT-Manager: It is VT (Visiting Time) controller with respect to each visitor at each heritage. Time difference between arrival of a visitor around a shelf where a given heritage is placed and his/her leaves yields the heritage VT (Visiting Time) of the visitor. The visiting time parameter is crucial in Question Generation Phase (for more details refer section 4.5.2).

Context Knowledgebase: It stores permanently static context from visitor profile. It also holds dynamic contexts temporarily during visiting session of each visitor.

4.2 Model Diagram of PCMGS

Out of the components discussed in previous section (Section 4.1) only research components in our work will be depicted in the model diagram (Fig 4.2). Preference Analyzer and Question Handler are the two components where the proposed model in this study involves. Preference Analyzer inputs proposed heritage for a given visitor and analyzes if those heritages are really the interest of the visitor and outputs the resulting heritage list (it a subset of the proposed heritage). Question Handler is to generate questions on behalf of the visitor customized as to the visitor profile so that the visitor may ask PCMGS by selecting from those questions. Each will be discussed in detail by the succeeding sections (Section 4.5 and Section 4.6)

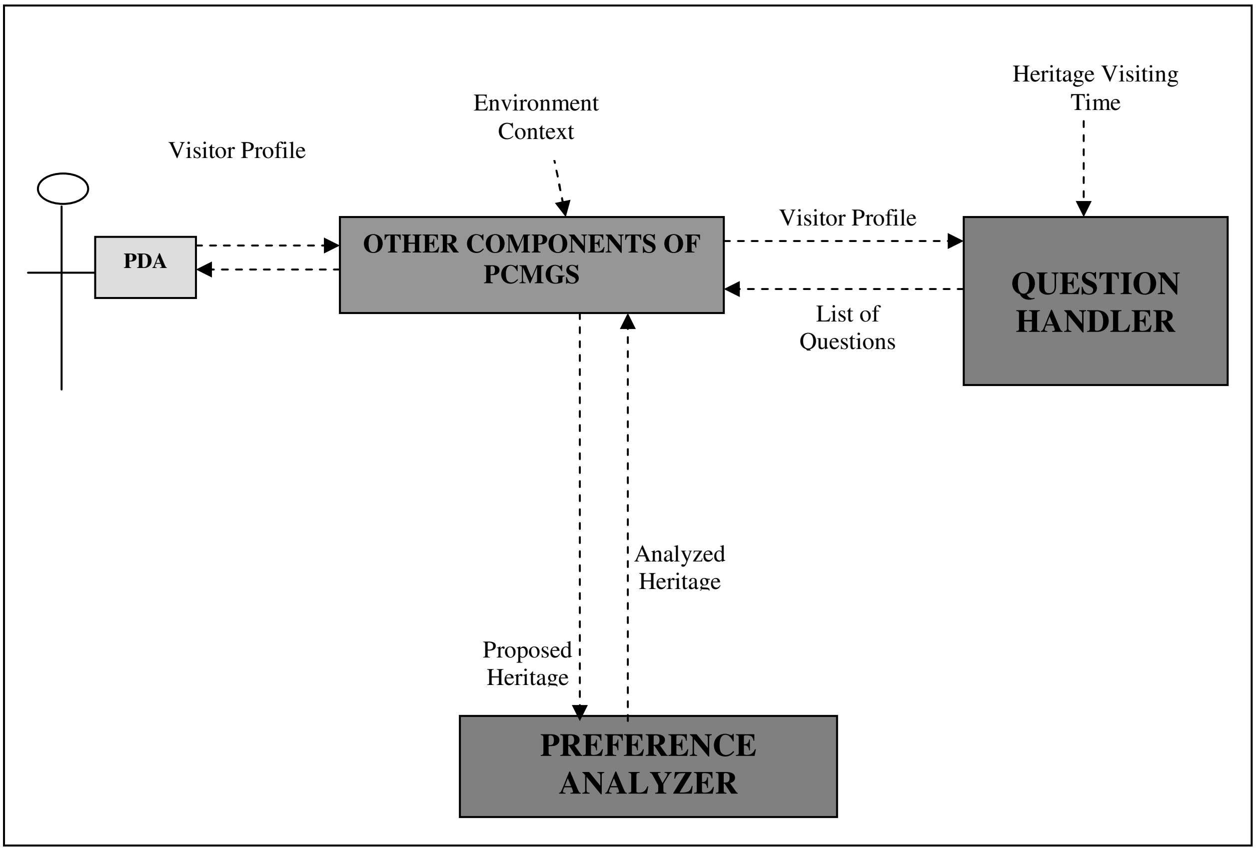


Fig 4. 2: Model Diagram of PCMGS

4.3 Flow of Events in PCMGS with Activity diagram

The flow of events in the proposed architecture of PCMGS is shown below in fig 4.3.

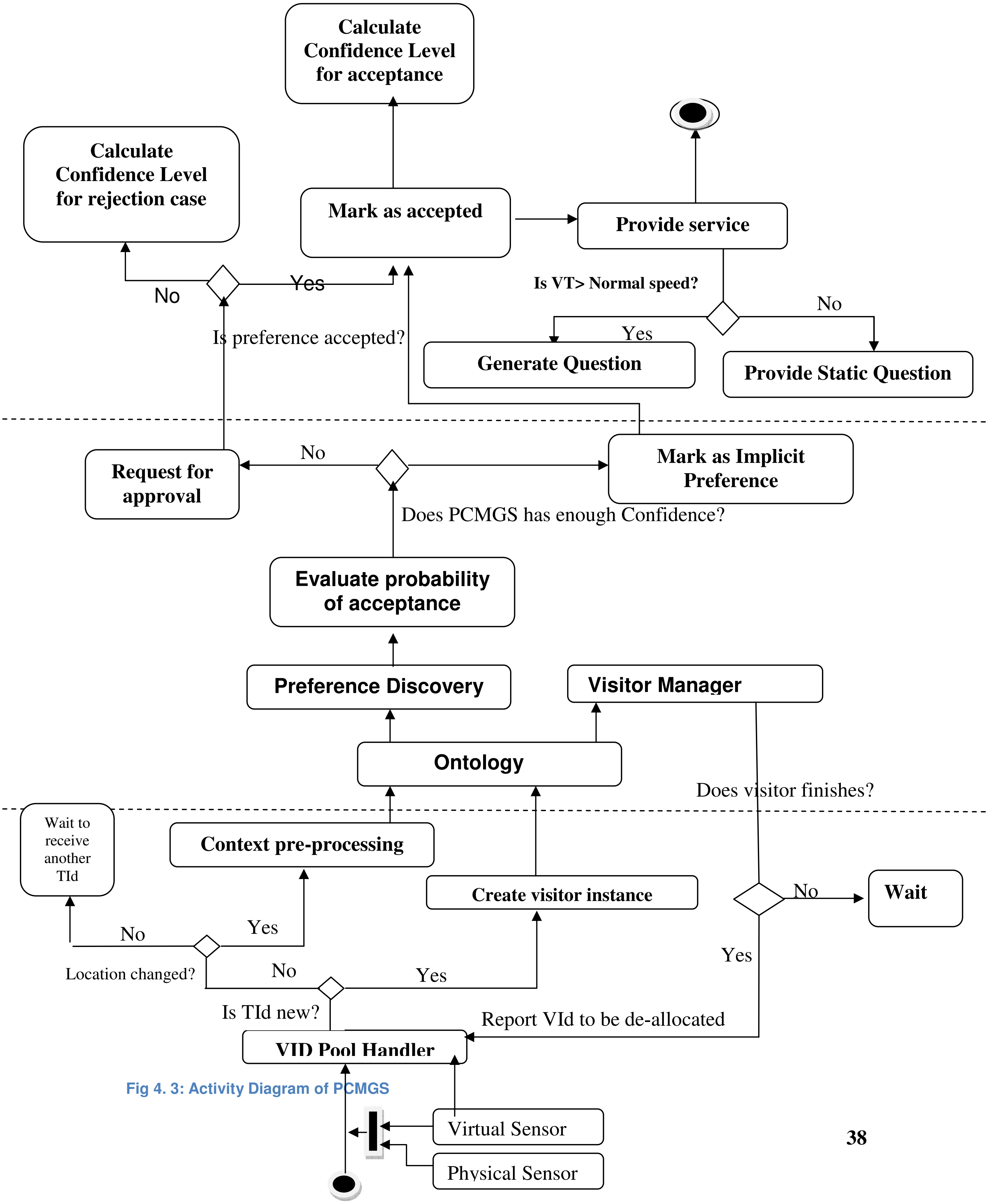


Fig 4. 3: Activity Diagram of PCMGS

4.4 General Scenario

The following table shows scenario that dictates general activities a given visitor may pass through while using PCMGS deployed on museum.

- A visitor is found in the main gate of National Museum of Ethiopia and holds PDA on which PCMGS is deployed.
- The PCMGS organizes static contexts (The visitor's profile from his PDA and/or other environment context if any) and dynamic contexts needed to know list of tour services for the visitor.
- The PCMGS discovers new preferences that may result in widening or narrowing Visitor's preference (new heritage preference which was not in preference list).
- The PCMGS checks current value of Confidence Level associated with each newly discovered preference. Then the higher the value of the confidence level, the lower the tendency to interrupt the visitor for discovered preference approval.
- Based on the acceptance or rejection of the discovered preference, the corresponding Confidence Level value is updated for future use.
- Then, the visitor is given with the site map of the heritages corresponding to his list of preferences and starts visiting.
- The visitor is now coming close to shelf 02 in room 03; Then PCMGS provides description of each heritage found on the shelf together with list of possible questions that can be asked for each heritage.
- The visitor continues visiting to the next room (room 04). Reading the description given for heritage 002 on shelf 01, he stays longer time than the normal visiting speed determined by PCMGS. Therefore PCMGS takes the assumption that the visitor faces something vague on the description and dynamically integrates the visitor's profile to generate list of questions which are different from the questions which are generated if the visitor were in normal visiting speed.
- The visitor selects the questions generated by PCMGS on behalf of his concern one by one and read the explaining answers accordingly.
- Then finishes the remaining heritages in normal visiting speed as to the time he scheduled to spend in the museum.

4.5 Details of Research Components in PCMGS

PCMGS has two phases, where the research components in this thesis work spread in to, guide service preparation phase and question handling phase. The phases are divided based on the guide service time. Phase one is before the guide service starts and phase two is after the visiting session starts. The first phase is dedicated to prepare context-aware guide service on individual visitor basis. The second phase comes after the visitor starts visiting and it may get triggered at any point in time of visiting through interaction due to questions. The detail of each phase is discussed in detail in the next section (Section 4.5.1 and Section 4.5.2 respectively).

4.5.1 Service Preparation Phase

The guide service rendered in PCMGS is not the same for every visitor rather customized service as to the interest of each visitor. In other words, all visitors joining the PCMGS service will not be rendered with the guide service over the whole heritages available rather subset of the available heritages. This is to increase quality of museum guide service in terms of providing adapted guide service in favour of visitor's interest. It demands ability of providing different service priority for different users and applications on their accord. Our approach leans to raising the context-awareness level of guide services provided to clients who are being served in pervasive environments. We consider three views at service discovery level. The first view takes decisive context data which are given higher priority during guide service discovery. View 2 is incorporating non functional (helping or additional) context data to extend view 1 and the last view is compressing view 1 due to non-negotiable context data. The foundation of each view bases on dynamic and static context data extracted from sensors and client profile.

View 1: Responding as to vital visitor context data such as heritage type preference.

View 2: Discovering new visitor preferences then recommend widening visitor's requirement, such as having relaxed time to have additional heritages to the preferences extracted on view 1 (extending client' preference).

View 3: Discovering new preferences then recommend narrowing visitor's requirement, such as having tight schedule not to finish the preferences extracted on view 1 (compressing client's preference).

View 1 is perfectly implicit and it doesn’t need any incorporation of recommendation approval for the service provided. However it is integrated with other two views (View 2 and 3) that incorporate recommendation to indirectly react to high-level user profile or derived contexts. View 2 is extending the proposed services in view 1 and view 3 is omitting some of candidate services selected for the user. Yet the newly proposed list of services in view 2 and 3 may or may not be liked by the user. Not to irritate the user by rendering guide services which are not of his/her interest, recommendation dispatching for the discovered preference is a very straight solution to get approval first. Approval request needs alarming of users to comment on the discovered preference prior to delivery of corresponding service. However this strategy is found working with the expense of higher user interruption to result in service dissatisfaction or even getting upset. To reconcile this conflict, our model deals with retrieval of corresponding services with minimal or zero approval request interruption.

4.5.1.1 Major Parameters used in Service Preparation phase of PCMGS

There are three major parameters that are used in derivation of Equation 1 namely Confidence Level, Negative Factor, Comment, and Confidence Level Parameter (α). Each of these parameters are described precisely as follows:

Confidence Level (CL): Represents the extent to which a recommendation will be accepted. Generally the higher the Confidence Level value, the greater the possibility of the discovered preference acceptance. This parameter is a key to make services implicit opposing to user destruction for approval. It is a data source associated with each preference discovered. It tells the level of confidence PCMGS has on the associated preference acceptance. For every discovered unique preference there is exactly one CL value. Therefore a set of CL value is mapped one to one with a set of preference or heritage instance.

Confidence Level Parameter (α): is a constant used as adjustment parameter over the previous Confidence Level value. It is added and subtracted from CLp for the case of recommendation acceptance and rejection respectively.

Negative Factor (NF): is a data source associated with each visitor. It holds the existing negative factors associated with the visitor not to accept a given preference approval request. It is accessed during CL building process for appreciating or compromising the constant value (α) that is used to update CL. For every visitor there is exactly one NF value. Therefore a set of NF value is mapped one to one with a set of visitors. Negative Factors are collected from visitors and/or PCMGS environment.

Negative Factor (NF) measures the negative situations of the user that contribute to reject a specific recommendation. It is a data source maintained by considering the existing negative context data over the PCMGS environment and visitor. The basic purpose of Negative Factor (NF) is to have knowledge source about external negative contexts that are involving when the user decides to accept or reject a given preference recommendation proposed from the pervasive environment. Negative factors are used to compromise rejection and appreciate acceptance. For instance, the acceptance of recommendation while negative factor exists leads to build a higher Confidence Level over the future acceptance of that recommendation than the case where there is no negative factor. The rejection of recommendation while negative factor exists leads to compromise the deducted value over the previous Confidence Level. Rejection followed by zero Negative Factor leads to nothing to compromise over the deducted value on Previous Confidence Level.

For example: Take a visitor in a situation of having tight schedule, sensitive to suffocation, non-conducive museum environment not air-conditioned, internal temperature is getting higher and higher, and number of visitors in the museum is increasing. Here PCMGS will collect two NFs.

CoMment (CM): CoMment (CM) used to represent user suggestion about a given heritage. It is a data source designed to be integer data type of three possible values to abstract appreciation, neutralism (nonalignment), and dissatisfaction. It is therefore associated with each heritage found in the museum. For every heritage there is exactly one CM value. Hence, a set of CM value is mapped one to one with a set of heritages.

Fig 4.4 depicts the interaction of the aforementioned parameters through their respective manger components. It also shows detail sub-components in Preference Analyzer component.

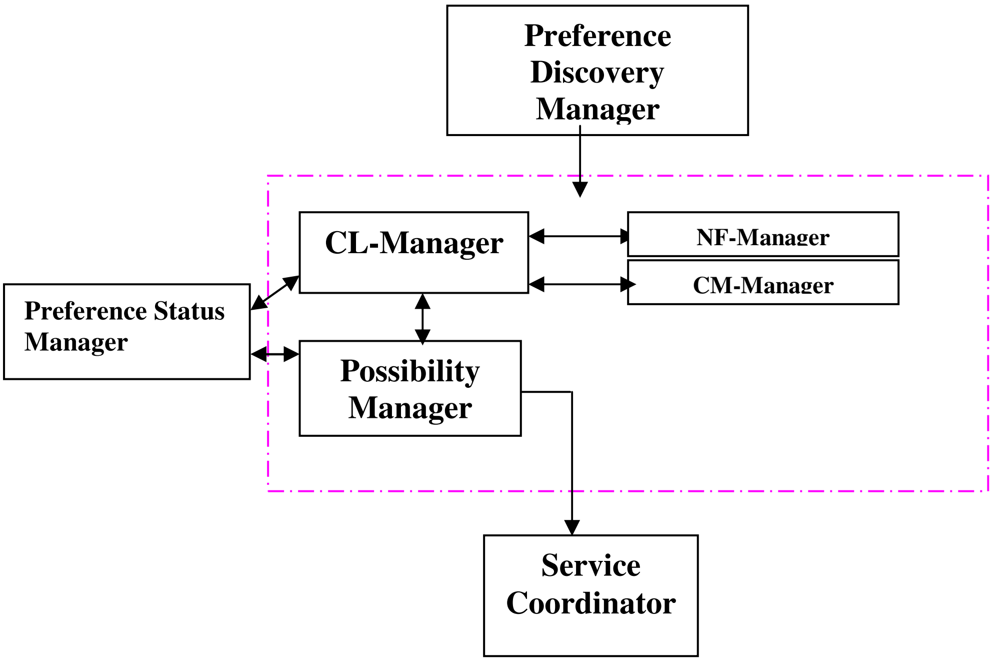


Fig 4. 4: Detail of Preference Analyzer Component

4.5.1.2 Derivation of Confidence Level (CL) Value

The process of learning (building Confidence Level over) possibility of acceptance of a particular preference passes through two cases. It is under conditions where recommendation is accepted and rejected respectively. There are additional parameters other than CL and NF, namely α (Confidence Level parameter), and CLp (Previous Confidence Level). See Equation 1 and read its immediate description that follows.

Recommendation Acceptance Case:

$$CL = CLp + CM + \alpha + NF$$

Recommendation Rejection Case:

$$CL = CLp - \alpha + NF$$

NB:
CL: Confidence Level
CLp: Previous Confidence Level
NF: Negative Factor
CM: CoMment
α: Confidence Level parameter

Equation 4. 1: Calculation of Confidence Level

The formula depicted in Equation 1 describes how the Confidence Level value is built or updated during learning phase. PCMGS must pass decision on whether the visitor has to be interrupted for preference approval or not. The decision process needs checking Confidence Level to get probability of acceptance (the probability in which the newly discovered preference will be accepted by the visitor). The derivation of CL value is discussed as follows.

Recommendation Acceptance Case: Increases the previous Confidence Level by a constant value of Confidence Level parameter (α). This is one case where CL value is updated. Acceptance lets PCMGS build a higher CL value over the previous CL value (CLp +α). Negative Factor on the visitor or around in PCMGS environment is considered to boost the current CL value as (CLp + α + NF). If there is no negative factor, only constant value, α is incremented (current CL value is assigned to CLp + α+ 0). If there is negative factor, the incremented value α is promoted by the value of NF (current CL value is assigned to CLp + α + NF). Finally, after enjoying the service of accepted preference, Comment (CM) value is collected which lets the calculation of current Confidence Level to be CLp + CM+α+NF.

Recommendation Rejection Case: This is the second case where CL value is updated. Rejection lets PCMGS build a lower CL value over the previous CL value (CLp -α). Negative Factor on the visitor or around in PCMGS environment is considered to moderate the current CL value as (CLp -α+ NF). If there is no negative factor, only constant value, α is decremented (current CL value is assigned to CLp - α+ 0). If there is Negative Factor, the decremented value α is compromised by the value of NF (current CL value is assigned to CLp - α + NF). Finally, there is no Comment (CM) value collected here because the visitor will not have the chance to visit the corresponding heritage to the discovered preference as it is rejected. Therefore the current CL value is assigned to expression CLp + 0 -α+ NF.

In both cases, Confidence Level is calculated as addition of CM, NF, and α parameters. This is because all of those parameters are independent of each other. The calculation is done each time after a user gets served and before the next access of CL reaches.

4.5.1.3 Assigning Values to Parameters

In this section, reason over values selected for each parameter found in calculation of Confidence Level is discussed.

Confidence Level (CL) is measured out of hundred percent, so that the minimum CL value to be zero and hundred is maximum. Any negative result for CL is adapted to zero and result greater than hundred is adapted to hundred. In general, the higher the CL value is, the greater the probability it contributes to preference acceptance.

Confidence Level Parameter (α) is a constant with value ten. This is to play with (increment or decrement) ten percent of the total Confidence Level each time acceptance or rejection of preference is encountered. We have chosen ten to provide moderate Confidence Level building period. In best case situation where every visitor with zero Negative Factor requested to approve a particular discovered preference accepts, it takes ten shot to build full confidence over future acceptance of this particular preference even with zero Comment value. In a case of optimal situation, where some visitors accept and some others reject a preference approval request, it takes even more shots in building full confidence than ten. In worst case, where every visitor rejects a request leads to continuous degrading of Confidence Level, which is to be re-arranged to zero Confidence Level at most and not negative. This is because the value of CL is bounded between zero and hundred.

Negative Factor (NF) is context to be counted as number of negative situations around a visitor and PCMGS environment. Therefore it is clear that its minimum value is zero (when there is nothing represented as negative situation). However we have made its maximum boundary to be $\alpha/2$. We are convinced that maximum value of NF must be less than Confidence Level parameter (α), because NF basically is introduced to appreciate or compromise the value added or subtracted (α) over the previous Confidence Level as to acceptance or rejection respectively. It is just adjustment parameter to reconcile the generally updated CL value with actual situation of the user or environment. Therefore it must not exceed the normally incremented or decremented value (α). We have selected $\alpha/2$, which is less than α, to be its maximum value. This allows to have moderate-paced learning scheme during acceptance and not to make lazy-fare reduction over the degraded Confidence Level during rejection.

Therefore its value can range between zero and $\alpha/2$. Meaning, if the number of negative factor does not exceed $\alpha/2$ it will be taken as it is during calculation of CL and the value $\alpha/2$ is taken if the number of negative factor exceeds $\alpha/2$.

CoMment (CM) is subjective and qualitative means of extracting user's perception over a given service. In our model CM is a parameter represented in a quantitative way to capture user's perception and it is used in evaluation of CL value. Although there are other detail representations of CoMment, we have generalized as appreciating (positive), not interested (negative) and non-aligned (neutral). Appreciation is represented as $\alpha/2$. Negative is represented as $-\alpha/2$ and Neutral as zero. Any value out of these discrete values will get rounded to the nearest among the three. Like Negative Factor, Comment is adjustment parameter to contribute additional information over updated CL value. Therefore it should not exceed the Confidence Level parameter and is best suited to be positive $\alpha/2$ or negative $\alpha/2$ for the two extreme comments. There will not be CM value for calculation of CL value in the case of discovered preference rejection; this is because if the user rejects a given recommended service, there is nothing to comment on.

4.5.1.4 Derivation of Probability of Acceptance

Once CL is calculated and stored for future access, it is retrieved whenever its associating preference is discovered and its corresponding service is to be prepared. The Probability of acceptance ($\mathbf{P_a}$) is the possibility a user will accept the proposed preference and is directly proportional to CL value. It is calculated as CL divided by hundred (The relative Confidence Level value out of hundred). The value of CL is between zero and hundred. Hundred is when the current value of CL reaches hundred or greater after a series of update process. Therefore $\mathbf{P_a}$ ranges from zero to one (as it is probability value).

$$\mathbf{P_a(preference) = CL\ (preference)\ /100}$$

NB:
P_a: Probability of acceptance
CL(preference): Confidence Level of a particular preference

Equation 4. 2: Calculation of Probability of Acceptance

The cut off point for $\mathbf{P_a}$ value is determined by the visitor profile. We have considered three visitor types that can be derived from the static visitor context or profile. Free visitors are relaxed visitors in their nature, optimists are those who are intermediate and pessimists are those who are conservative. PCMGS will read the three qualitative visitor types from PDA and maps them into quantitative values to use in mathematical processes. Visitor types with their associated $\mathbf{P_a}$ cut off point is shown in the following table (Table 4.1).

Table 4. 1: Types of Visitors vs. Pa Cut-off Points

Type of visitor	**P_a Cut-off Point**	**Remark**
Free visitor	0.60	If CL is greater than 60%
Optimist	0.75	If CL is greater than 75%
Pessimist	0.90	If CL is greater than 90%

Probability of acceptance or P_a value greater than cut off point implies that the preference will be accepted by the user therefore there is no need to interrupt the user for preference approval. Probability of acceptance value less than cut-off point leads to interruption of visitor. If approval is to be requested for a particular preference (PCMGS is not sure about the preference if it is to be accepted by the visitor or not) the CL value is updated immediately after request is answered. The parameters participate in updating of CL value are Negative Factor (NF), CoMment (CM) and Confidence Level manipulation parameter (α). However, once PCMGS builds enough Confidence Level over this preference, there is no need to request for approval. Yet CL value should be updated right after the associating guide service with this preference is rendered. Here the only parameter to influence calculation of CL is CoMment (CM). Positive comment appreciates the CL value, zero comment leaves it intact, and negative comment degrades it back to the time of learning or building Confidence Level again. Therefore there are ways to degrade a Confidence Level over the preference after it reaches its maxima that keeps the dynamicity of our mode to the varying context of visitors and PCMGS environment. Fig 4.5 depicts when does each parameter is important during calculation of CL. As we can see from the figure there are three parameters involving on calculation of

CL until it reaches its maxima for each visitor type. After reaching the maxima of each visitors there is only one parameter to affect CL value.

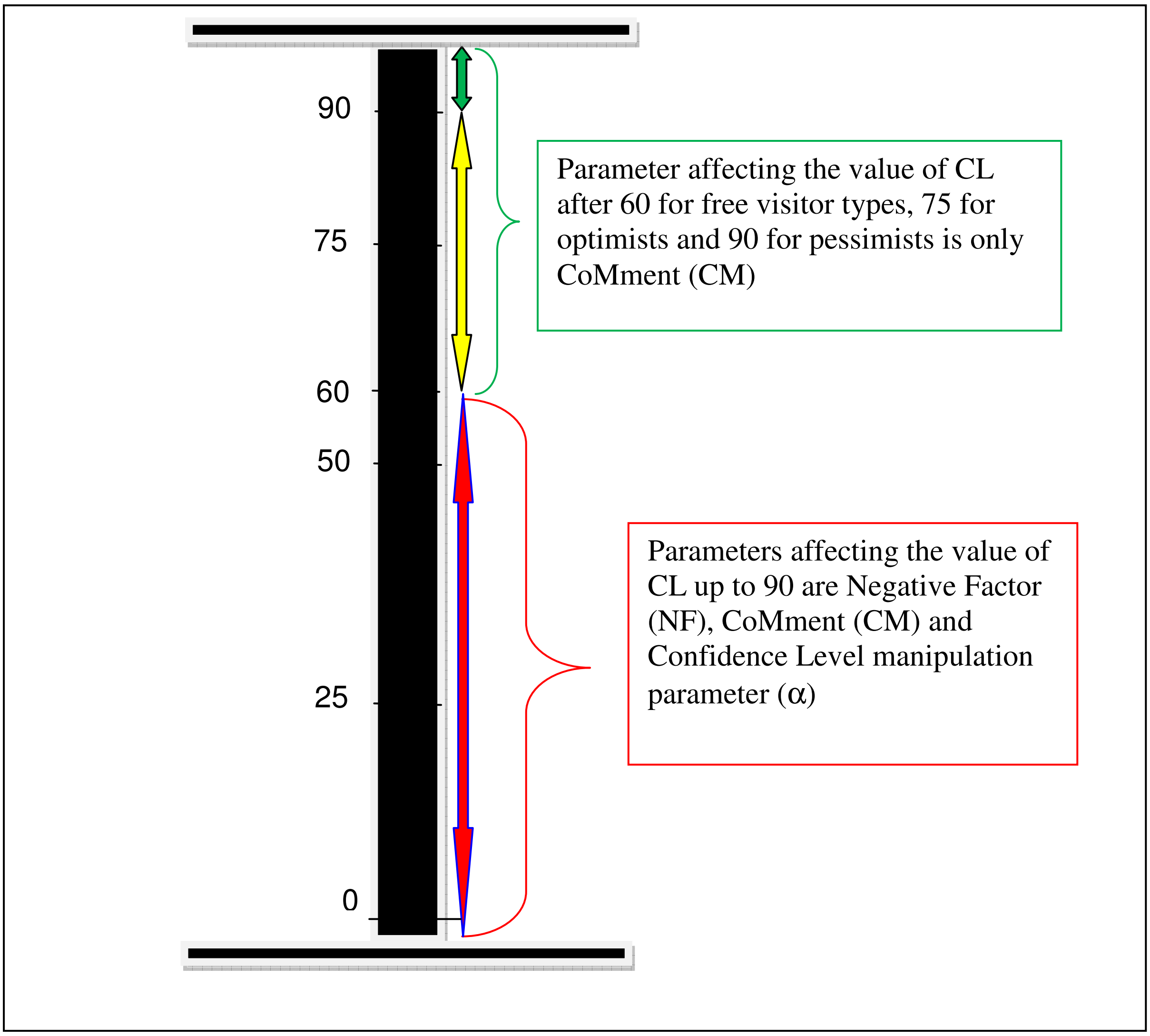


Fig 4. 5: Parameters Involvement at Different Levels of CL values

The procedure for the proposed approach in Service Preparation phase is outlined as follows:

- Discovering available service with respect to user preference. This is proposing different service for different user, based on the user preference that is pre-processed to include some preferences and/or omit some others.
- Prior to service rendering, there has to be preference approval process that is decided to done be implicitly or explicitly. This decision process is supported by PCMGS's Confidence Level over the discovered preference. The CL value consulted is either

zero or already passes through a series of arithmetic manipulations during previous recommendation. It is evaluated as follows:

- Rate the discovered preference in terms of the possibility to be accepted by the user. The possibility of acceptance is directly proportional to the available Confidence Level learned over a specific preference. The higher the rate the more likely to be accepted by the user that implies the higher the rate the better it is not to interrupt the user (implicitly consider it as the original visitor's preference).
- Acceptance of recommendation plays a positive role in raising value of Confidence Level. It increments previous value of confidence level by α(The Confidence Level manipulation parameter). If Negative Factor (NF) exists, CL gets another value, NF, to add up. This is because acceptance of a given recommendation while there are Negative Factor that retards the acceptance leads to appreciate value of Confidence Level. If there is no negative factor, there is nothing to promote over the normal added value α.
- Rejection of recommendation put negative influence on Confidence Level value. It decrements the previous value of Confidence Level by α. If Negative Factor avails, the decrement operation should be reconciled by adding a value NF. This is because rejection of a given recommendation while there are negative factors that facilitate rejection directs to compromising the deducted value over the Confidence Level. If there is no negative factor, there is nothing to compromise.
- Spotted CoMments (CM) play significant role in the possibility of recommendation acceptance. CM is maintained while the visitors spotted on heritages. The higher the number of positive comments supports to have greater Confidence Level (CL) value.

4.5.1.5 Example on Service Preparation Phase: Calculation of CL value

To demonstrate the ideas discussed in preceding sections, (especially service preparation phase), we have considered the situations where PCMGS interacts with its user. Bellow is the example with list of activities:

- *David is from Canada and he is 32. He is professor in Cultural evolutionary science and his heritage category preference is that of cultural heritages, he understands only English, he is pessimist in nature, volunteer to see already visited heritages in other museums, the museum is not air conditioned and it is getting hot inside, he wants to stay in the museum only for 1hour.*

- *Out of the David's profile, PCMGS finds out that there are 2 Negative Factors (the museum is not air conditioned and getting hot inside and he has tight schedule).*
- *After organizing David's profile, PCMGS discovers guide service groups A which contains heritages 0 up to 9. PCMGS already built enough Confidence Level over the first 8 heritages (CL value is greater than 90 and David is pessimist) therefore adds them automatically into the accepted service map list. However for the 8th and 9th heritages it requests David for approval. David accepts the 8th and rejects 9th. Then he visits the heritage 0 up to 8 (total of nine heritages) in English descriptions coming to his PDA. Finally David is asked to provide comment and passes-on positive comment for all of the heritages he has visited.*
- *Then PCMGS updates the value of CL for all of the proposed guide services for David as follows. Since the first 8 preferences (heritage 0 up to 7) are automatic, their respective CL value is updated by the Comment value David provides. For the 8th preference (Heritage 9) the respective CL value is updated by adding α and Negative Factor, and Comment provided. For the 9th preference (10th heritage) the respective CL value is updated by subtracting* α, adding Negative Factor and CoMment (zero).
- *Sosina reaches while David finishes and get disconnected from PCMGS. She has similar visitor profile as David, yet she is optimist in nature and has relaxed time with no boundary.*
- *PCMGS discovers service groups A and C, heritage 0 up to 9 and heritage 20 up to 29 (a total of 20 heritages selected for her).*
- *PCMGS has already built enough Confidence Level over the first 16 heritages and the 19th heritage (their respective Confidence Level is greater than 75 and Sosina is optimist).*
- *Sosina is requested to approve only the 26th, 27th, and 28th heritages. She rejects all of them, and provided with service map with 17 heritages.*
- *PCMGS calculates the first 16 heritages and the 19th Confidence Level with the comment she provides.*
- *However, the requested three heritages' (26th, 27th, and 28th) Confidence Level is calculated with the Negative Factor, CoMment (Zero, and Confidence Level manipulation parameter (α).*
- *Sosina finishes and exit PCMGS.*

4.5.2 Question Handling Phase

Different visitors take different visiting time for the same heritage because of various contexts of the environment and/or the visitor. We have noticed that there are two alternatives to assign the time taken to visit each heritage. The first option is to assign the time stamp in advance for each heritage statically. The second way is to assign dynamic visiting time for each heritage based on the visitor's profile. Therefore Visiting Time (VT) is a data source to hold visiting time value for a given heritage. For every heritage there is exactly one attached VT value. While the visitor is visiting a given heritage s/he may finish with in or out of the predefined visiting session.

The Question Generation phase is the stage that comes after the visitor finishes reading the short heritage description provided on her/his PDA. Every heritage may not be fully explained in a short description provided in on shot. Therefore further detail is given through questions the visitor may raise. In our model the visitors are not supposed to provide their question on their word (The model does not support natural language processing to understand visitors' question). The way visitors provide their question is by selecting from list of questions provided by the model itself. Here our model prepares question on behalf of the visitor and provides the list. Whenever the visitor selects a question from the list and submits, the answer will be provided again to the visitor.

Each heritage has its own list of static (fixed) questions which are to be provided for visitors who are assumed to understand the heritage description more. If the visitor is assumed not to understand the description there should be dynamic question generator on behalf of the visitor. Question Generation Phase is to generate question lists as to the context of the visitor. It considers the visiting speed of the visitor on a particular heritage in terms of VT parameter. Then the visitor will ask PCMGS by selecting from the list of questions generated. There are two possibilities where the visitor uses the pre-defined visiting session. They may be in-time, on-time, or over-time.

1. If the visitor uses a predefined VT value (finishes in time or on time), PCMGS will provide static list of questions in the assumption that the visitor understands the short description provided. Therefore the visitor sees the list and can select his questions if any.

2. If the visitor exceeds the limit of pre-defined VT value (Stays longer time than the assigned VT value on the shelf that contains the heritage), PCMGS will provide

dynamic list of questions in the assumption that the visitor does not understand the short description displayed on his/her device.

In this phase, the term static refers to the concept ‘Being the same for all visitors’. The question is prepared in the same manner for every visitor in advance and is to be displayed to the visitor device along with heritage description. The term dynamic refers to the concept ‘Being different for different visitors’. The dynamic question generation module bases on visitor’s profile such as educational background, skill, experience, nationality, age, and language preference. That is questions will be prepared by considering visitors profile.

4.6 Museum Ontology Design and Dynamic System Model

UML is a diagrammatic language for expressing the kind of software design that is known as object-oriented design. It helps to use diagrams for modeling objects in a system. It provides a standard notation for the modeling of a real-world object as a first step in developing an object-oriented design methodology. Diagrams used in UML such as sequence diagram, activity diagram, class diagram, and so on illustrate interacting objects within a given process model [38].

Ontology is a common vocabulary for describing the concepts that exist in an area of knowledge and the relationships that exist between them. Ontology allows having a more detailed specification of the relationships in particular domain knowledge than is the case with taxonomy. The resulting vocabulary can be used by computers as well as understood by humans. It defines or specifies the concepts, relationships, and other distinctions that are relevant for modeling a domain [39].

An abstract or general idea inferred from specific instances or collection of instances is referred to as Concept. Ontology concepts are, therefore, general entities interacting in ontology knowledgebase of a model. Ontology concepts can be extracted from UML representation [35].

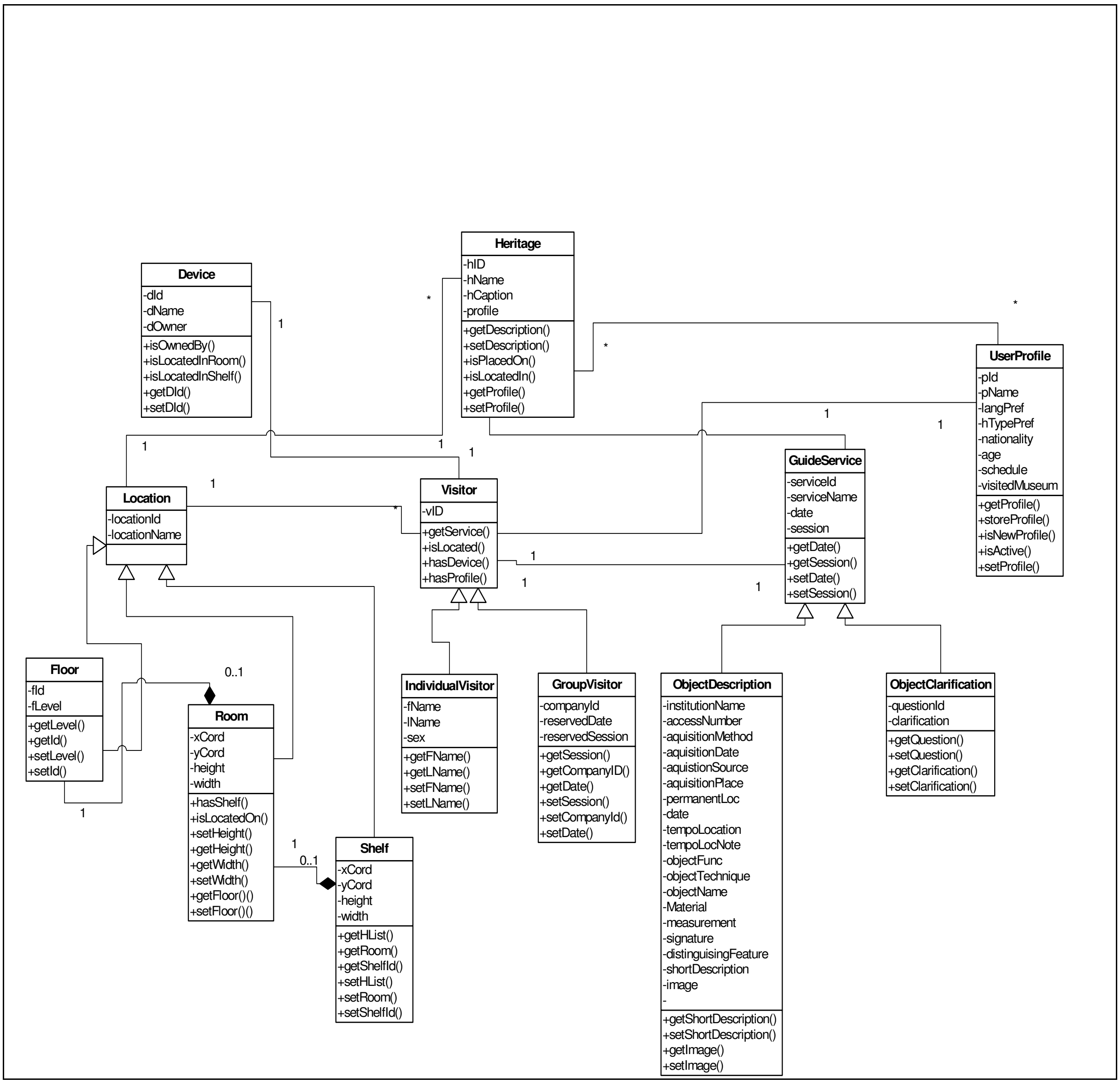


Fig 4. 6: Class Diagram of PCMGS

Fig 4.6 depicts class diagram in UML representation of top level PCMGS classes' that dictates high-level objects and their relationship around the pervasive context aware museum guide environment. The dynamic model is also depicted in sequence diagram on fig 4.7.

From our class diagram and sequence diagram we get initial constructs to start mapping museum ontology. Classes in the class diagram are mapped into ontology concept/classes, attributes are mapped into data and state.

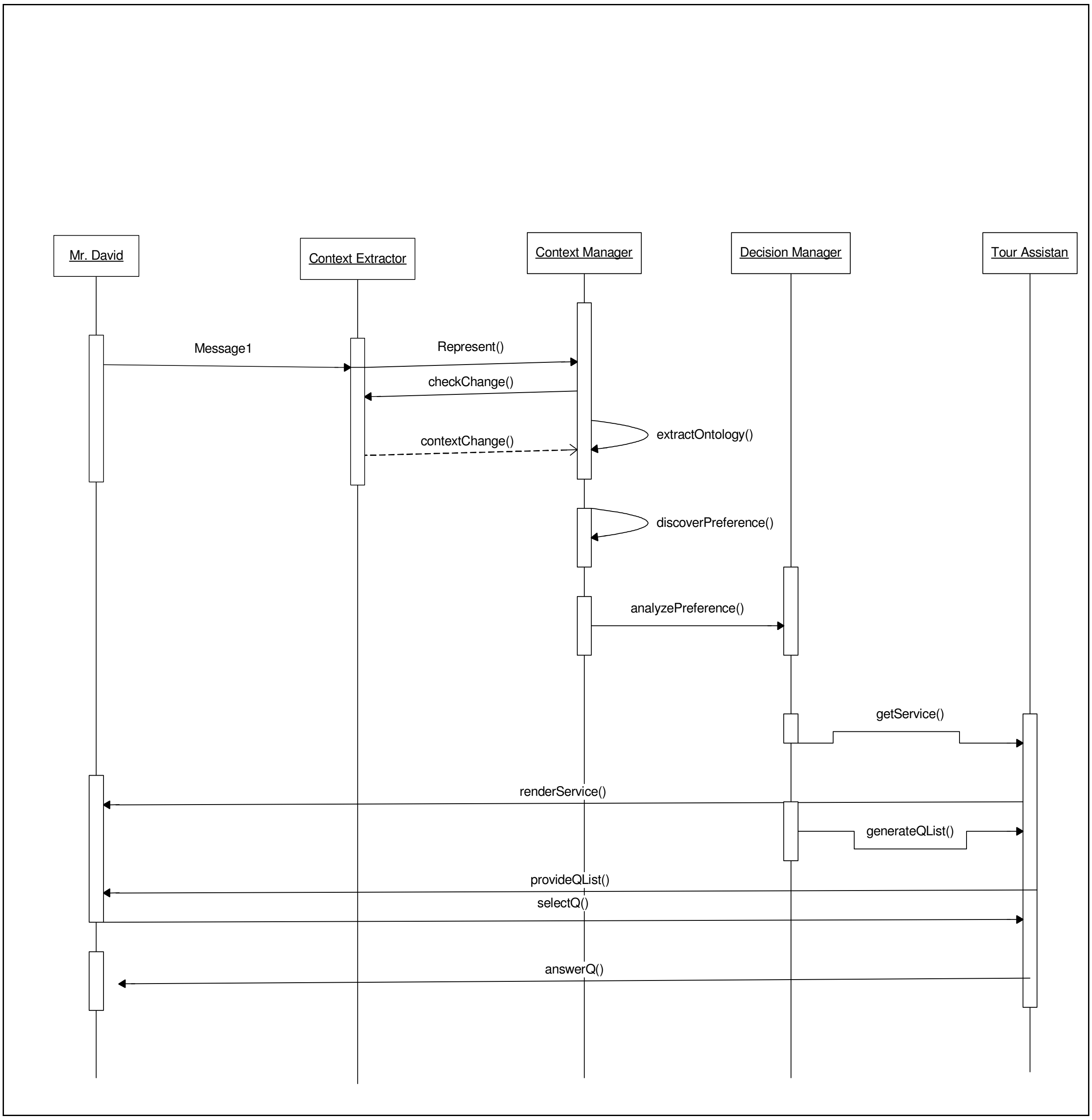


Fig 4. 7: Sequence Diagram of PCMGS

4.6.1 Concepts on Our Museum Ontology

Ontology concepts are classes whose interactions represent a system. We have identified the following ontology concepts. There are 14 classes of which Museum_Entity is a super-class of the remaining 13. Table 4.2 illustrates ontology classes in museum ontology of PCMGS.

Table 4. 2: PCMGS Ontology Concepts

No	Concept	Description	Inherits\Is sub-class of
1	Museum_Entity	Root class of museum ontology which all remaining classes derived from. It is an abstract representation of any object interacting in PCMGS around pervasive museum environment.	Thing (Super class of every ontology concept)
2	Visitor	A person who joins PCMGS service for the purpose of visiting heritages in the pervasive environment.	Musuem_Entity
3	Group_Visitors	Group of visitors joining PCMGS service at once. Visitors from schools and organizations.	Visitor
4	Heritage	An object placed in shelf of museum, exhibited for visitors.	Musuem_Entity
5	Preference	Guide Service: List of heritage descriptions for selected preferences\heritages to a particular visitor.	Musuem_Entity
6	Guide_Service	Abstract representation of visitor preference with respect to a given heritage. It is designed to be an entity that holds attributes and	Musuem_Entity

		behaviours of set of guide service objects.	
7	Location		Musuem_Entity
8	Room	A rectangular place found in Museum site area. It contains shelves.	Location
9	Shelf	A rectangular place found in Room. It contains heritage.	Location
10	Device	Any equipment in PCMGS pervasive environment.	Musuem_Entity
11	RFID_Tag	A radio frequency sensitive tag given to visitors when first joins PCMGS. It is carried by visitors so that their location can be tracked.	Device
12	RFID_Reader	A device to track RFID_Tags	Device
13	PDA	A device carried by visitors as Personal Digital Assistant	Device
14	Question	Any Question raised by Visitors.	Musuem_Entity

4.6.2 List of Properties in Our Museum Ontology

Properties let ontology designers to show the behaviour of identified concepts. There are different property types of which Object Property and Data type Property are some. Object property basically represents the relationship between objects. Data type property is a class representing attributes. Each property has its own domain and range. Domain and range of object properties are from the user-defined classes. Out of ontology properties we have identified, Table 4.3 depicts only major object properties with their domain, range and inverse relationship across classes. Out of the defined classes in Table 4.2, major data type properties of few classes, namely Guide_Service, Visitor, Heritage, and Preference are listed in Table 4.4.

Tabel 4. 3: PCMGS Ontology Object Properties

No	Property Name	Domain	Range	Inverse of
1	Asks	Visitor	Question	askedBy
2	Comments	Visitor	Heritage	
3	containsHeritage	Shelf	Heritage	containedByShelf
4	containedByShelf	Heritage	Shelf	containsHeritage
5	discoveredPreferen ce	Visitor	Preference	
6	has Accepted	Visitor	Preference	
7	hasRejected	Visitor	Preference	
8	hasAssociatedPrefe rence	Heritage	Preference	hasAssociatedHeri tage
9	hasAssociatedHerit age	Preference	Heritage	hasAssociatedPref erence
10	isVisitingHeritage	Visitor	Heritage	isBeingVisitedBy
11	isBeingVisitedBy	Heritage	Visitor	isVisitingHeritage
12	isAskedAbout	Question	Heritage	
13	isAskedBy	Question	Visitor	asks
14	isAnsweredTo	Question	Visitor	
15	isStandingInfrontOf	Visitor	Shelf	
16	isPreparedToVisitor	Guide_Service	Visitor	isAssistedByGuide ServiceInstance
17	Owns	Visitor	Device	isOwnedBy
18	isOwnedBy	Device	Visitor	owns
19	trackedTagPoint	RFID_Tag	Location	
20	hasNetPreference	Visitor	Preference	
21	isAssistedByGuide ServiceInstance	Visitor	Guide_Service	isPreparedToVisit or
22	isLocatedIn	Visior	Room	

Tabel 4. 4: PCMGS Ontology Data type Properties

No	Property Name	Domain	Range	Description
1	correspondingVId	Guide_Service	String	Visitor's identification number, who served by the guide instance
2	nextServiceStatus	Guide_Service	Integer	Next service map number
3	Serviced	Guide_Service	String	Service identification number
4	firstName	Visitor	String	Visitor's first name
5	visitorId	Visitor	String	Visitor's identification number
6	finishesVisitingSession	Visitor	Boolean	Tells whether visitor finishes or not
7	visitingSpeed	Visitor	Integer	Visiting speed in which the visitor is visiting a given heritage
8	visitingStartTime	Visitor	DateTime	The time at which the visitor starts visiting
9	Age	Visitor	Integer	Visitor's age
10	isNewCommer	Visitor	Boolean	Checks if visitor is new comer or not
11	alreadyVisitedMusuem	Visitor	Boolean	Checks if visitor

				already visits other museums or not
12	languagePreference	Visitor	String	The language preference of the visitor
13	nativeCountry	Visitor	String	The native country visitor comes from
14	hasNegativeFactor	Visitor	Integer	Counts the number of negative factors around visitors and/or PCMGS
15	timeToSpentHere	Visitor	DateTime	Visitor's schedule
16	accessNumber	Heritage	String	Access number in which the heritage is identified
17	acquistionSource	Heritage	String	Source of heritage
18	acquistionDate	Heritage	DateTime	The date
19	estimatedVSpeed	Heritage	Integer	Estimated visiting speed attached to each heritage
20	commentValue	Heritage	Integer	Comment marked on a given heritage
21	heritageType	Heritage	String	Heritage category
22	shortDescriptionAmharicVersion	Heritage	String	Short description given to visitors about a given heritage in Amharic
23	shortDescriptionEnglishVersion	Heritage	String	Short description given to visitors about a given heritage in English
24	pId	Preference	String	Preference

				identification number
25	confidenceLevel	Preference	Double	Confidence level value on a given preference

Fig 4.8 also represents objects with their hierarchical chain using ontology graph. Fig 4.9 shows the concepts in Museum Ontology along with their property. Fig 4.10 describes sample instances of Room and Shelf Concepts in the Ontology.

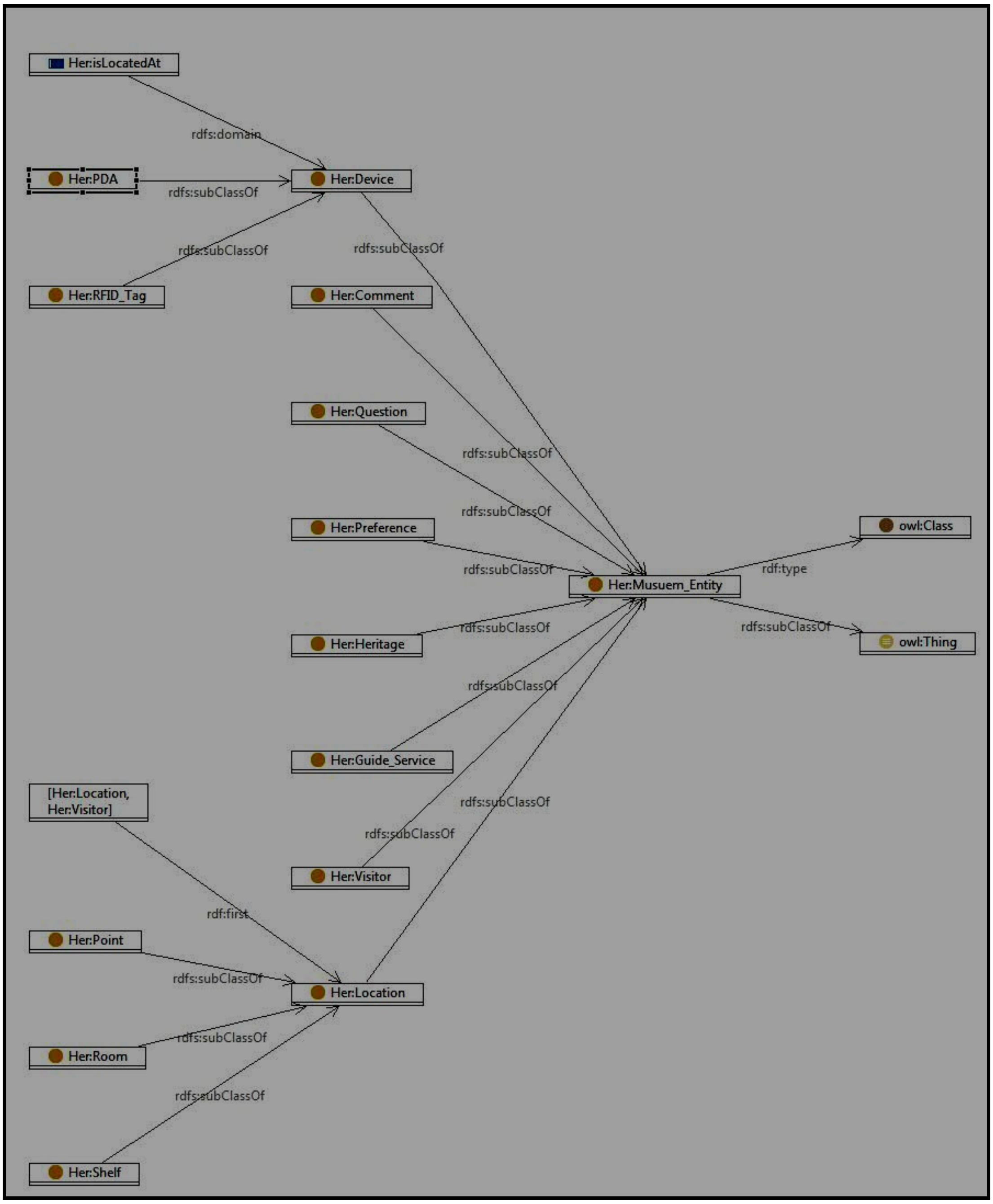
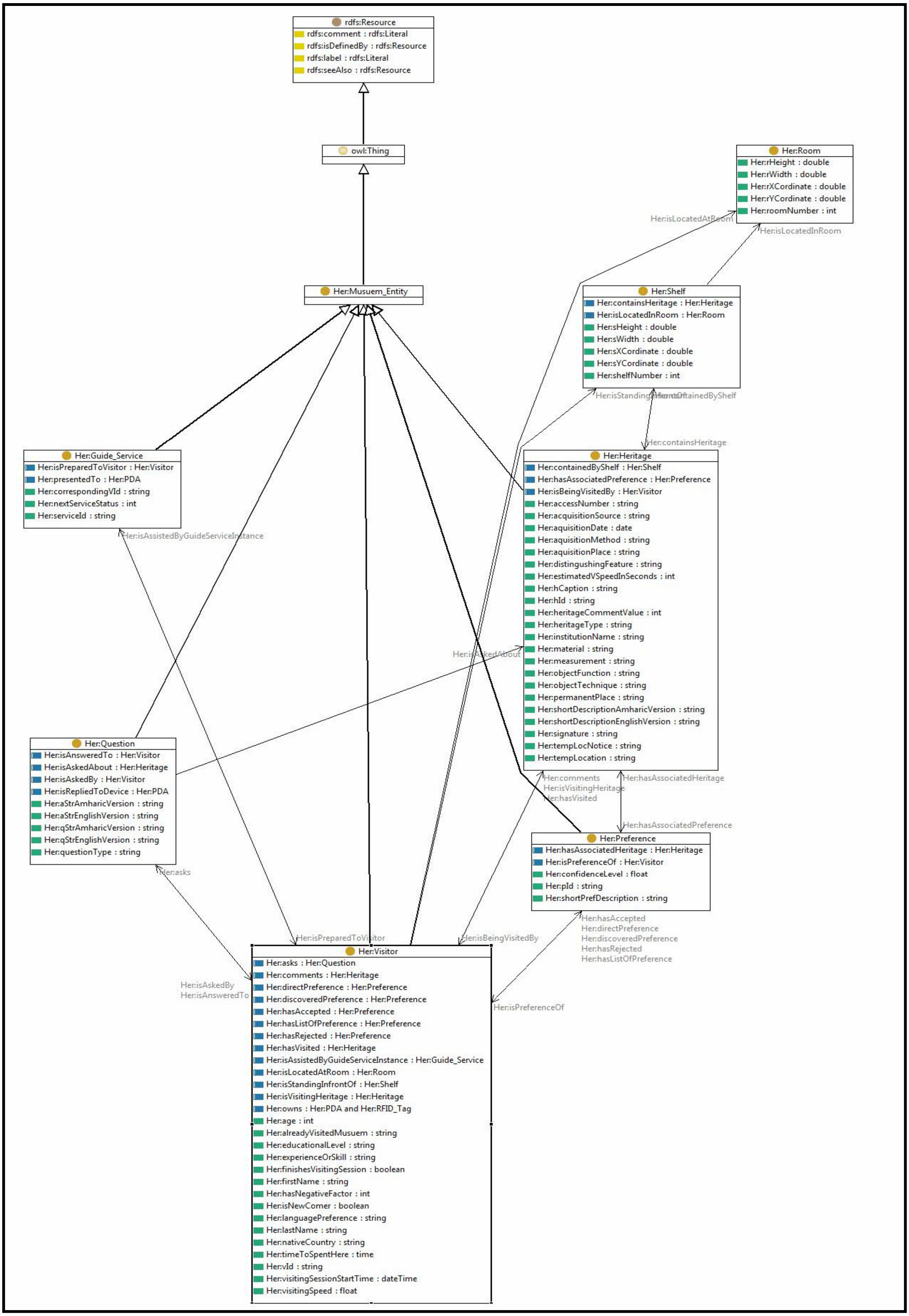


Fig 4. 8: PCMGS Ontology Concept Hierarchy with Graph



rdfs:Resource
rdfs:comment : rdfs:Literal
rdfs:isDefinedBy : rdfs:Resource
rdfs:label : rdfs:Literal
rdfs:seeAlso : rdfs:Resource
owl:Thing
Her:Musuem_Entity
Her:Room
Her:rHeight : double
Her:rWidth : double
Her:rXCordinate : double
Her:rYCordinate : double
Her:roomNumber : int
Her:isLocatedAtRoom
Her:isLocatedInRoom
Her:Shelf
Her:containsHeritage : Her:Heritage
Her:isLocatedInRoom : Her:Room
Her:sHeight : double
Her:sWidth : double
Her:sXCordinate : double
Her:sYCordinate : double
Her:shelfNumber : int
Her:containsHeritage
Her:Guide_Service
Her:isPreparedToVisitor : Her:Visitor
Her:presentedTo : Her:PDA
Her:correspondingVId : string
Her:nextServiceStatus : int
Her:serviceId : string
Her:isAssistedByGuideServiceInstance
Her:Heritage
Her:containedByShelf : Her:Shelf
Her:hasAssociatedPreference : Her:Preference
Her:isBeingVisitedBy : Her:Visitor
Her:accessNumber : string
Her:acquisitionSource : string
Her:aquisitionDate : date
Her:aquisitionMethod : string
Her:aquisitionPlace : string
Her:distingushingFeature : string
Her:estimatedVSpeedInSeconds : int
Her:hCaption : string
Her:hId : string
Her:heritageCommentValue : int
Her:heritageType : string
Her:institutionName : string
Her:material : string
Her:measurement : string
Her:objectFunction : string
Her:objectTechnique : string
Her:permanentPlace : string
Her:shortDescriptionAmharicVersion : string
Her:shortDescriptionEnglishVersion : string
Her:signature : string
Her:templocNotice : string
Her:tempLocation : string
Her:isAskedAbout
Her:Question
Her:isAnsweredTo : Her:Visitor
Her:isAskedAbout : Her:Heritage
Her:isAskedBy : Her:Visitor
Her:isRepliedToDevice : Her:PDA
Her:aStrAmharicVersion : string
Her:aStrEnglishVersion : string
Her:qStrAmharicVersion : string
Her:qStrEnglishVersion : string
Her:questionType : string
Her:asks
Her:comments
Her:isVisitingHeritage
Her:hasVisited
Her:hasAssociatedHeritage
Her:hasAssociatedPreference
Her:Preference
Her:hasAssociatedHeritage : Her:Heritage
Her:isPreferenceOf : Her:Visitor
Her:confidenceLevel : float
Her:pId : string
Her:shortPrefDescription : string
Her:hasAccepted
Her:directPreference
Her:discoveredPreference
Her:hasRejected
Her:hasListOfPreference
Her:isPreparedToVisitor
Her:isBeingVisitedBy
Her:isAskedBy
Her:isAnsweredTo
Her:isPreferenceOf
Her:Visitor
Her:asks : Her:Question
Her:comments : Her:Heritage
Her:directPreference : Her:Preference
Her:discoveredPreference : Her:Preference
Her:hasAccepted : Her:Preference
Her:hasListOfPreference : Her:Preference
Her:hasRejected : Her:Preference
Her:hasVisited : Her:Heritage
Her:isAssistedByGuideServiceInstance : Her:Guide_Service
Her:isLocatedAtRoom : Her:Room
Her:isStandingInfrontOf : Her:Shelf
Her:isVisitingHeritage : Her:Heritage
Her:owns : Her:PDA and Her:RFID_Tag
Her:age : int
Her:alreadyVisitedMusuem : string
Her:educationalLevel : string
Her:experienceOrSkill : string
Her:finishesVisitingSession : boolean
Her:firstName : string
Her:hasNegativeFactor : int
Her:isNewComer : boolean
Her:languagePreference : string
Her:lastName : string
Her:nativeCountry : string
Her:timeToSpentHere : time
Her:vId : string
Her:visitingSessionStartTime : dateTime
Her:visitingSpeed : float

Fig 4. 9: PCMGS Ontology Class Relation

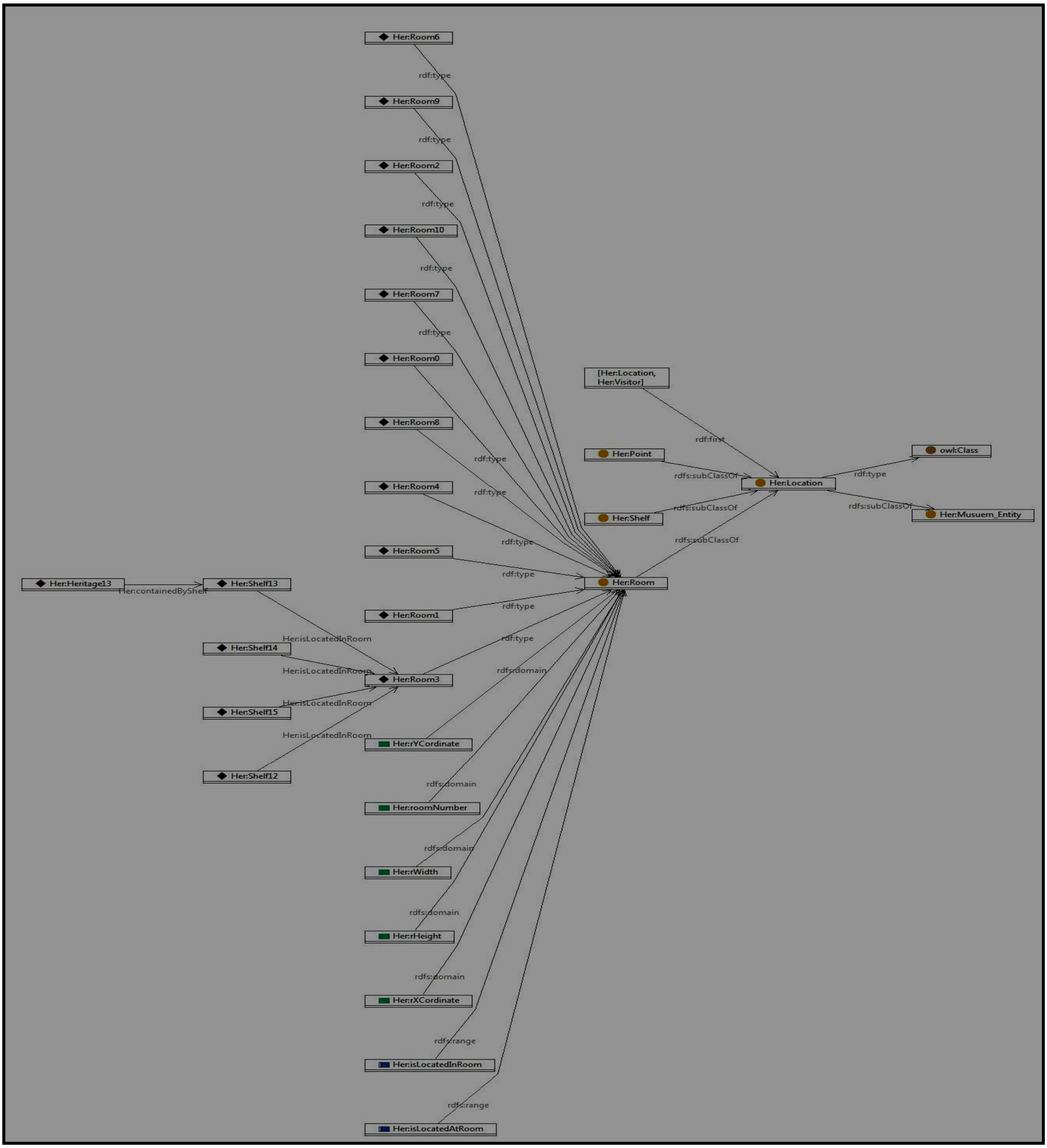


Fig 4. 10: PCMGS Ontology Instances from Room and Shelf Classes

CHAPTER 5: IMPLEMENTATION

In this chapter prototype implementation detail of PCMGS will be discussed. The chapter is organized into four sections from which the first (Section 5.1) provides list of tools and technologies utilized; section 5.2 will give implementation detail of each layers in proposed architecture and algorithmic description of major modules there. User defined rules are described in section 5.3. Section 5.4 is to show implementation scenario and demonstration will be on section 5.5.

5.1 Implementation Detail

In this section, implementation detail of major modules is discussed as to their architectural layer.

5.1.1 Context Sensing Layer

This layer contains two modules, context sensor module and context pre-processor module. The first take responsibility of sensing location of visitor in terms of X-Y coordinates. The second pre-processes the raw context into high level context information.

Real implementation of this layer needs physical devices such as RFID readers, RFID tags, and Software Sensors. However, the prototype implementation of our research work uses simulation of these physical devices. By developing Java threads that provides random location-identity pair continuously while the PCMGS prototype is running, we are able to simulate RFID readers and tags. This simulator generates random identity-location pair by taking assumption of visitor movement from one shelf to other and from one room to another room. The identity element of location-identity pair is produced to represent TId (Tag Identification) and later pre-processed to be used as Visitor Id. Location is random X-Y coordinate based point to imitate pattern of human being (visitor in our case) movement.

Here we have assumed to have straight arrangements of rooms and shelves. Each room has length of 8 meters and width 4 meters. There are 4 shelves straight at each room with 2 meters, and width 1 meter. Therefore total area coverage of the museum is to be 80 meters by 4 meters. We have also assumed to have 100 RFID tags to be given to visitors. Algorithm of RFID readers is depicted in the algorithm 5.1.

Algorithm 5. 1: Context Sensor Module

1. Create tlHash to store visitorId-visitorLocation a hashmap with (key-value) pair
2. xCordinate (random number between [0,1] x79) +1
3. yCordinate ⟵ (random number between [0,1] x79) +1
4. point⟵ new point object(xCordinate, yCordinate)
5. tagId⟵(random number between [0,1] x99) +1
6. tlHash⟵ (tagId, point)
7. return tlHash

Once the physical context of the visitor is sensed as raw context from virtual sensors and hardware sensors simulator, here comes task of context data pre-processing to feed input to the next layer, context management layer. This module is aggregation of three sub-modules namely, VID Pool, VID Pool Handler, and Context Pre-processor. VID Pool sub-module is a Java method storing TId (Tag Identification) and location using hash map so that every visitor identification number is acting as a unique key and respective location point as value, VID Pool Handler sub-module is a Java method to control the VID Pool in that it filters the random raw location generated. It checks if the identity-location pair, is new or not. If it is new, it is added to the VID Pool otherwise it checks again if the location is changed or not, if the location is changed, the new location will be registered otherwise wait for another random location to be generated. Context-Pre-processor sub-module is Java class that maps the random location point in to logical context information such as room number or shelf number. It also pre-processes TId (Tag Identification) into VId (Visitor Identification) by just concatenating the string "Visitor" into the TId. Algorithm of context pre-processing is depicted in algorithm 5.2.

5.1.2 Context Analysis Layer

At the layer of context analysis there are about__ major modules, namely ontology manager, preference discovery, preference analyzer, and preference status manager. The ontology manager contains Java methods that enable to create ontology class instances, edit instance property values, and access values.

Preference discovery module is a Java method that uses Weka API to classify and discover heritage preference based on knowledgebase constructed dynamically from previous visitors. This module uses the J48 algorithm that inputs learning, which is prepared from previous visitors' profile stored in file format of .arff. It also accepts another file with the same file format containing profile of the newly coming visitor for whom the decision is to be made. Algorithm of preference discovery is depicted in algorithm 5.3.

Algorithm 5. 2: Context Pre-processor Module

```
1. Input:
        R (set of room objects having height 8 meters and width 4 meters)
        S (set of shelf objects having height 2 meters and width 1meter)
        VID-pool (hash map containing list of visitorId-visitorLocation pair)
2. Trigger context-sensor module
3. Get VID-pool
4. Get random key-value hash of id- currlocation pair
5. If(id is found in VID-pool)
    a. If(currlocation value of id is same as prevlocation)
        i. Do nothing
        ii. wait
    b. Else
        i. prevlocation =currlocation
        ii. Return id- location pair
6. Else
    a. Register the new visitor into the pool
    b. Return id-location pair
7. End If
8. Extract location from id
9. For every ri element of R
    a. If(ri contains currlocation)
        i. roomIndex ← i
        ii. go to step 10
        iii. end if
    b. increment i
    c. go to step 9
10. For every si element of S
11. For every si element of S
    a. If(si contains currlocation)
        i. shelfIndex ← i
        ii. go to step 11
        iii. end if
```

b. increment i
c. go to step 10
11 If prevShelfIndex=shelfIndex then visitor is not moving
12 Else
13 Return roomIndex
14 Return shelfIndex

Algorithm 5. 3: Preference Discovery Module

1. Input:
   pvProfile (dynamic data source with list of visitor profile and respective service)
   nvProfile (new visitor profile in terms of age, educational background, heritage type preference, visitor type, already visited museum, schedule, language preference)
2. Output:
   Dsg (discovered service group)
3. Read pvProfile
4. Create Instance of weka API as training data set
5. Read nvProfile
6. Create Instance of weka API as testing data set
7. Pre-process instances
8. Call J48 algorithm to build decision tree for the instance to be tested
9. Predict the dsg for nvProfile
10. Get dsg
11. Map dsg into heritage, index
12. For every ds element of dsg,
13. If (dsg is group A)
    a. For each i, starting from 0 to 10
       H (index) ⟵ heritage (i)
    Return H
14. Else If (dsg is group B)
    a. For each i, starting from 10 to 20
       H(index) ⟵heritage(i)
15. Else If (dsg is group C)
    a. For each i, starting from 20 to 30
       H(index) ⟵heritage(i)
16 Else If (dsg is group D)
    b. For each i, starting from 30 to 40
       H(index) ⟵heritage(i)
17 Else If (dsg is group all)
    c. For each i, starting from 0 to 39
       H(index) ⟵heritage(i)

```
18 increment index
19 go to step 13
20 Return H
```

Preference analyzer is Java class having, NF-Manager, CM Manager, CL-Manager, and Possibility Manager as instance methods. Preference status manager is a Java class to manage and update preference with respect to each heritage. Algorithm of preference analyzer is depicted in algorithm 5.4.

Algorithm 5. 4: Preference Analyzer Module

```
1. Input:
     dp (set of discovered preference for the visitor whose preference is to be
analyzed)
     Visitor (User whose discovered preference is to be analyzed)
2. Output: ap (set of analyzed preferences resulted after analysis)
3. possibiligyOfAcceptance            false
4. For each dpi in dp
        a. Get confidenceLevelValue(dpi)
             i. If(confidenceLevelValue>=0.6 and visitorType="Free")
                     PossibiligyOfAcceptance ←— true
            ii. If(confidenceLevelValue>=0.75 and visitorType="Optimist")
                     PossibiligyOfAcceptance ←— true
           iii. If(confidenceLevelValue>=0.90 and visitorType="Pessimist")
                     PossibiligyOfAcceptance ←— true
            iv. End if
        b. if(possibilityOfAcceptance= "true")
             i. api=dpi
        c. else
             i. Request for approval from the visitor
            ii. If(visitor accepts di)
                   1. api=dpi
           iii. Else go to step call calculatePreference module with input
                Visitor, dpi, comment=0, acceptanceFlag=false

5. Return api
```

Preference analyzer model uses a crucial parameter in this research work, Confidence Level. The algorithm of CL calculation is depicted in algorithm 5.5.

5.1.3 Service Layer

There are three major modules here, service coordinator, service progress manager and service provider. Service coordinator is to handle random arrival of visitors and call appropriate methods that lets visitor join PCMGS. Service provider extracts the heritage description as per service parameters handed from the service coordinator module. Algorithm of service coordinator and service provider is depicted in algorithm 5.6 and 5.7 respectively.

Algorithm 5. 5: Calculation of CL

1. Input:
   Preference
   Visitor
   Comment
   AcceptanceFlag
2. previousCLValue ⟵ get confidenceLevel from Preference
3. negativeFactor ⟵ get negative factors from Visitor
4. α, confidence level parameter, =10
5. If(AcceptanceFlag=”true”)
   CurrentCLValue ⟵ previousCLValue + α + negativeFactor + comment
6. Else
   CurrentCLValue ⟵ previousCLValue - α + negativeFactor + comment
7. If CurrentCLValue less than or equal to zero
   a. CurrentCLValue ⟵ 0
8. Else if CurrentCLValue greater than or equal to 1
   a. CurrentCLValue ⟵ 1
9. Set confidenceLevel of Preference to currentCLValue

Algorithm 5. 6: Service Coordinator Module

1. Input
   ActiveVisitorIdPool (set of active visitors id pool)
2. visitorId ⟵ random and unique visitor Id
3. If (activeVisitorIdPool contains visitorId)
   a. Sense location of visitor from sensor module
   b. Call context-preprocessor module
   c. Call service pool handler module
4. Else
   a. Call preference discovery module
   b. Call preference analyzer module
   c. Add to activeVisitorIdPool
   d. Call service pool handler module
5. Go to step 1

Algorithm 5. 7: Service Provider Module

1. Input:
    Visitor
    currRoomNumber
    currShelfNumber

2. For each counter, from 0 to 9
3. If currRoomNumber=counter
    a. Indexer=0
    b. If currShelfNumber=Indexer
        i. If visitor prefers Amharic
            Send Amharic version of heritage description on room(index), shelf(index) into visitor's PDA
        ii. Else if visitor prefers English
            Send Amharic version of heritage description on room(index), shelf(index) into visitor's PDA

    c. Else if currShelfNumber= Indexer+1
        i. If visitor prefers Amharic
            Send Amharic version of heritage description on room(index+1), shelf(index+1) into visitor's PDA
        ii. Else if visitor prefers English
            Send Amharic version of heritage description on room(index+1), shelf(index+1) into visitor's PDA

    d. Else if currShelfNumber= Indexer+2
        i. If visitor prefers Amharic
            Send Amharic version of heritage description on room(index+2), shelf(index+2) into visitor's PDA
        ii. Else if visitor prefers English
            Send Amharic version of heritage description on room(index+2), shelf(index+2) into visitor's PDA

    e. Else if currShelfNumber= Indexer+3
        i. If visitor prefers Amharic
            Send Amharic version of heritage description on room(index+3), shelf(index+3) into visitor's PDA
        ii. Else if visitor prefers English
            Send Amharic version of heritage description on room(index+3), shelf(index+3) into visitor's PDA

4. End if
5. Go to step 2

5.2 Ontology Rule

Among the user defined rule we have developed, this section discusses about the major ones. Table 5.1 demonstrate these rules along with their description.

Table 5. 1: PCMGS Ontology Rules

User-defined Rule	Description
[HeritagePlacementRule1: (?H Her:containedByShelf ?S)->(S? Her:containsHeritage ?H)]	If a given heritage H is placed in a shelf S, then that shelf contains the heritage H.
[HeritagePlacementRule2: (?H Her:containedByShelf ?S)(?S Her:isAtRoom ?R) ->(?H Her:isLocatedInRoom ?R)]	If a given heritage is contained by shelf S, and S is at room R, then H is located in room R.
[HeritageCommentRule1: (?V Her:comments ?H)->(?V Her:hasVisited ?H)]	If a visitor V comments on heritage H, than the visitor has already visited that Heritage H.
[HeritageCommentRule2: (?V Her:comments ?H) (?P Her:hasAssociatedHeritage ?H) ->(?V Her:hasAccepted ?P)]	If a visitor comments a given heritage. He/she was already accepted that heritage
[VisitorLocationRule1: (?V Her:isStandingInfrontOf ?S) (?V Her:owns ?D) (?S Her:containsHeritage ?H) (?G Her:isPreparedFrom ?H) ->(?V Her:isVisitingHeritage ?H)(?G Her:presentTo ?D)	If a visitor is standing in front of shelf S, carrying a device D, and if S, contains heritage H and a guide service (description) is prepared from H, then the visitor is visiting H and the guide service G is presented to D.
[VisitorLocationRule2: (?V Her:isStandingInfrontOf ?S) (?S Her:isLocatedInRoom ?R) ->(?V Her:isLocatedAtRoom ?R)]	If a visitor is standing in front of a shelf and that shelf is located at room R, therefore the visitor is located at room R.

[DeviceLocationRule1: (?D Her:ownedBy ?V)(?D Her:isLocatedAt ?L) ->(?V Her:vIsLocatedAt ?L)]	If a device is owned by a visitor V, and the device is located at location L, then the device is located at L.
[VisitorQuestionRule2: (?V Her:owns ?D) (?V Her:asks ?Q) (? Q Her: has Answer ?A) ->(?A Her:answeredTo ?V)]	If visitor V, owns device D, and asks question Q, and Q has answer A, therefore A is sent to D

5.3 Prototype Implementation and Scenario

5.3.1 Tools and Technologies Used for Prototype

Several tools and technologies were utilized for the purpose of developing the prototype of PCMGS. Programming, communication, database management, context representation, context reasoning and operating environment used in the prototype implementation are listed as follows:

- Java 2 Micro Edition (J2ME) version 2.5.1 for CLDC (Connected Limited Device Configuration) of the MIDP (Mobile Information Device Profile) is used for developing the mobile device edition of the prototype.
- Java 2 Standard Edition (J2SE) version 1.5.0_06 is used for developing the desktop machine edition of the prototype.
- TopBraid Composer Maestro Edition is used for developing Museum Ontology
- Jena Semantic Web framework version 2.4.2 is used for generating an RDF model and implementing the ontology and user-defined reasoning rules.
- Microsoft Windows XP operating system is used as the PCMGS server.
- Weka 3.7.0 API (Application Programming Interface) is used to build decision tree for the purpose of visitor preference discovery

5.3.2 **Implementation Scenario**

1. The day is Saturday and Rosa was very busy the whole morning at her office. She plans to enjoy the time after noon relaxing by visiting heritages at National Museum of Ethiopia (Where PCMGS is deployed) and go to cinema at 4:30 P.M with friends.
2. She reaches at the gate of the museum at 2:00 P.M having her PDA.
3. Wondering when PCMGS launches automatically as she passes the main gate, Rosa entertains the welcome message into her PDA. Then after, she is informed that her profile is to be loaded to the PCMGS server so as to discover heritages to be visited on behalf of her.
4. Rosa is very happy and let PCMGS to proceed. Then immediately 20 heritages are selected for Rosa with total visiting time not exceeding 2 hours because she has to leave the museum at 4:00P.M.
5. Out of the selected 20 heritages PCMGS is not sure whether Rosa will accept the 5^{th}, 9^{th}, 12^{th}, 16^{th}, and 19^{th} discovered preferences. This is because the Confidence Level over each of these preferences is less than the threshold value PCMGS expects. The remaining preferences are assured by PCMGS that Rosa will like them.
6. For the doubted preferences, PCMGS requests Rosa to approve her acceptance. Rosa accepts all of them except 16^{th} and 19^{th}.
7. PCMGS then prepares automatically service map (list of heritages id and name) containing 18 locations (room number and shelf number) where Rosa has to go around.
8. Rosa is now in room 2 and around shelf 3 where the first heritage in her service map is located. Triggered by the logical location of Rosa, PCMGS starts to send heritage description placed on room 2 and shelf 3 into Rosa's PDA. The description is given in Amharic because PCMGS reads her language skill to be Amharic from her PDA at the beginning.
9. Rosa finishes visiting heritage on room 2 and shelf 3; PCMGS accepts comment on the heritage she has visited and removes this heritage from the service map. Moving to around shelf 4, the same room, she has noticed that PCMGS is sending the current description found on shelf 4.
10. Rosa follows the service map provided to visit all heritages there and comment accordingly.

11. While she finishes the visiting session, she has been informed to be disconnected from PCMGS and leaves the museum.

5.4 Prototype Demonstration

When a visitor first joins PCMGS environment, the system deployed there checks if the visitor is newly joining or not. For new visitors, it displays the welcome window and starts to capture the visitor profile as a static context of the visitor. Fig 5.1 shows summarized visitor profile having seven context properties (among the context properties we proposed we have used only seven of them to test our model). These context elements are taken to determine at which class of service that visitor is classified into. The determination is done from the huge knowledge base of previous visitor profile (previous visitor history).

Fig 5 1:Visitor Profile Display

Analysis of visitor preference is done by checking PCMGS's Confidence Level over the acceptance of that preference with the available cut-off points. If PCMGS gets greater confidence level value than 0.6 for free visitors, 0.75 for optimist visitors and 0.90 for pessimist visitors, the heritage is taken as automatic and the visitor is not asked to approve. Only those preferences whose Confidence Level value less than the cut-off point will be presented for the visitor to decide if they have to be accepted or not. This is true if the checking comes in negative response (PCMGS is not sure about the acceptance of that heritage. Fig 5.2 shows user PDA, that lets the visitor provide his/her decision.

Fig 5 2: Preference Approval Display

Once the preference analysis is over, the visitor is given with list of selected heritages by their number so that the visitor will use this list as a proposed service map provider. The list is dynamically updated when the visitor continues visiting. Finally the visitor will be disconnected from PCMGS service after the list comes empty. Fig 5.3 depicts sample service map containing four heritages (Heritage23, Heritage24, Heritage28, and Heritage33).

Fig 5 3: Service Map Display

Fig 5.4 shows snap shot of heritage description while visitors are visiting. This display lets visitors to comment or inquire questions concerning the description given.

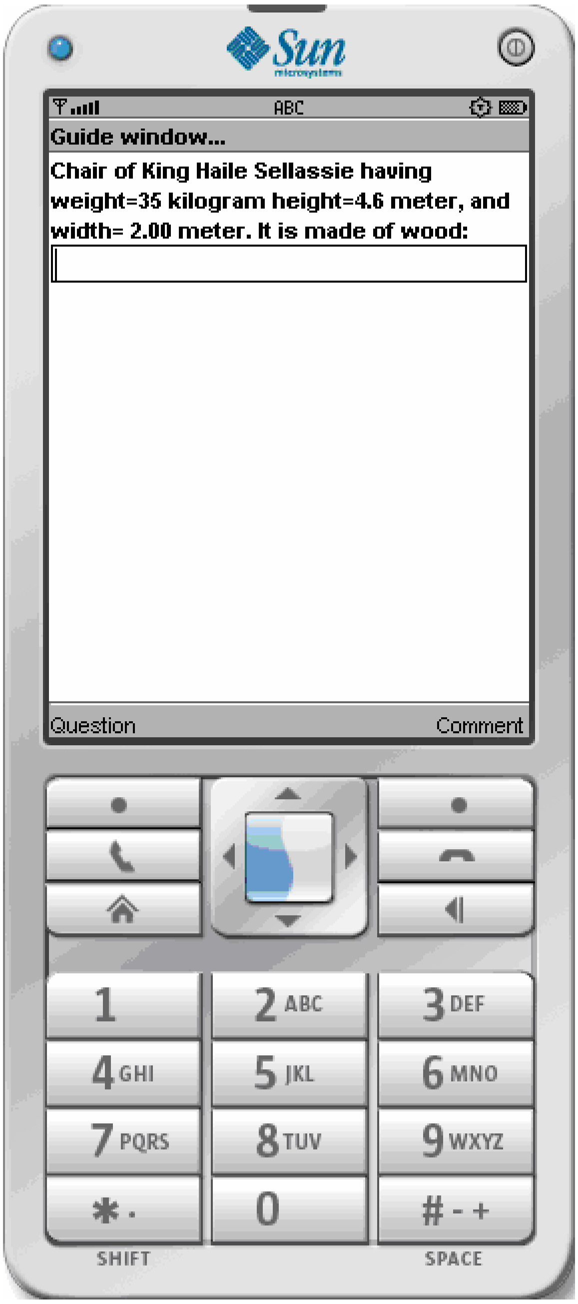


Fig 5 4: Heritage Description Display

Fig 5.5 demonstrates the commenting display that lets visitors enter their qualitative comments mapped into quantitative values. The display dictates the visitor to enter five if they are in deep appreciation, zero if they are fairly interested and negative five if they found the heritage they have visited not as they expect before. The three quantitative values (-5, 0, and 5) are used to map the visitor perceptions about the heritage.

Fig 5 5: Comment Display

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this thesis, we first identified the shortcomings with the existing pervasive context-aware museum guide models with regards to quality of museum guide service. On the course of solution modeling of the shortcomings, we have identified and addressed our research problems and made significant contribution to the pervasive museum environments.

6.2 Contribution of the Work

To address these shortcomings, we proposed a pervasive context-aware museum guide model. In the proposed architecture, we have made a research contribution to address the five issues as follows:

- Introduced a new approach to service users in pervasive environments with minimal or zero destruction. To find this promise working, we have introduced quantitative parameters to represent qualitative contexts and perceptions. Of these parameters, Confidence Level (CL) is the major one (as discussed in chapter 4) that represents the extent to which a proposed service is really liked by the user.
- Push-based approach followed in calculation of CL value increases the performance at preference analysis stage in terms of response time.
- Introduction of preparing customized questions as per the profile of the user comes up with a new approach to track whether a given visitor is in trouble or not. To accomplish this function, we proposed a parameter referred to as VT (Visiting Time). VT represents the estimated visiting time of each heritage under normal condition
- The dynamic nature of our knowledgebase manipulation lets the proposed model work with up-to-date situations (up-to-date user and environment contexts).
- Design of museum ontology which can be adapted and re-used to any application around tourism domain.

In order to demonstrate the validity of the proposed pervasive context-aware museum guide service architecture, we selected a pervasive museum scenario for which we developed a prototype implementation (explained in chapter 5). In the prototype, we have implemented the major components in Service Preparation phase of the proposed architecture.

6.3 Future Research Directions

Although we tried our best to realize the proposed architecture for pervasive museum guide systems under the objective of addressing the shortcomings of existing works, we do not believe that the architecture is generic enough to incorporate potential issues in pervasive museum guide systems. For instance, despite the importance of the issues, we have not considered the privacy and security aspect of context information in the course of all phases in our architecture since it was beyond the scope of this work. Therefore, we believe that the proposed architecture can be enhanced in such a way that the privacy and security of context is taken into account while visitors profile is to be extracted from their PDA.

The other line of improvement is regarding the complete implementation of the prototype. Even though we implemented many of the components in the proposed architecture, components in Question Generation Phase are not included in our prototype implementation. Moreover, the physical sensor we have simulated bases on X-Y coordinates and it is not capable of tracking visitors' location in terms of three dimensions. Hence, by taking this prototype as a platform, a more realistic implementation and deployment of the proposed architecture is another slot of improvement so as to maximize its usability in a real-life setting in the future

Incorporation of Question Answering Techniques so as to boost the quality of guide service in terms of maintaining the knowledge transferred to visitors. This is vital from the angle at which the perception of visitors about heritages visited is directly affected by the knowledge they incurred after the guide service. In order to describe the heritage as they are, may need a dialog between the visitor and guidance (human being or software guide services). Mostly this dialog is based on answering the questions asked by visitors. Therefore I strongly recommend future researchers to take this motivation, merging our model (PCMGS), and question answering techniques to make the museum guide service complete and more realistic.

Incorporation of social aspect of visitors around tourism opens another research direction. It is to let tourists share ideas and experiences around the pervasive environment they are enjoying.

REFERENCE

[1]. Mark Weiser, "The Computer for the 21st Century", Scientific American Journal, September 1991.

[2]. Dr. Seng Loke, "Context-aware pervasive systems architecture for a new breed of applications", Auerbach Publications Taylor & Francis Group, an informa business, New York.

[3]. Birhanu Mekuria, "Context information refinement for pervasive medical systems", Addis Ababa University, Department of Computer Science, Addis Ababa, 2007.

[4]. M. Satyanarayanan "Pervasive Computing: Vision and Challenges", Carnegie Mellon University, IEEE Personal Communications, August 2001

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

[6]. Dey, A.K. and Abowd, G.D, "Towards a better understanding of context and context-awareness", Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, New York, 2001.

[7]. Matthias Baldauf, Schahram Dustdar, Florian Rosenberg," A Survey on Context-Aware Systems", Distributed Systems Group, Information Systems Institute, Vienna University of Technology, Argentinierstrasse 8/184-11040 Vienna, Austria, International Journal of Ad Hoc and Ubiquitous Computing, Inderscience Publishers, January 2007.

[8]. Schilit, B. and Theimer, M, "Disseminating active map information to mobile hosts", IEEE Network, Vol. 8, No. 5, page 22–32, 1994.

[9]. Hull, R., Neaves, P. and Bedford-Roberts, J., "Towards situated computing", Proceedings of the First International Symposium on Wearable Computers (ISWC '97), p.146, 1997.

[10]. T. Gu et al., "A service-oriented middleware for building context-aware services", Journal of Network and Computer Applications 28, Page 1–18, 2005.

[11]. Feruzan Ay, "Context Modeling and Reasoning Using Ontology", University of Technology Berlin, 2007.

[12]. Ariel Pashtan, Remy Blattler, Andi Heuser, and Peter Scheuermann, "CATIS: A Context-Aware Tourist Information System", Dept. of ECE, Buffalo Grove

Northwestern University, IL 60089, USA Evanston, IL 60208, USA, IMC 2003, Proc. 4th International Workshop of Mobile Computing, Rostock, June 2003

[13]. Adriano Albertini et.al., "An Agent-Based Architecture for Museum Visitors' Guide Systems", Department of Management Information Systems University of Haifa, Haifa, Israel, and ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica, Povo, Trento, Italy, 2006.

[14]. Luca Buriano, "Exploiting Social Context Information in Context-Aware Mobile Tourism Guides", Telecom Italia - Innovation & Engineering Services. Reiss Romoli 274, 10148 Torino, Italy, 2005.

[15]. Shih-Chun Chou, Wen-Tai Hsieh, Fabien L. Gandon and Norman M. Sadeh, "Semantic Web Technologies for Context-Aware Museum Tour Guide Applications", Advanced e-Commerce Institute for Information Industry Taipei, Taiwan, ROC, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213-3891 – USA.

[16]. Gregory D. Abowd , Christopher G. Atkeson , Jason Hong , Sue Long, Rob Kooper , Mike Pinkerton, "Cyberguide: a mobile context-aware tour guide", Wireless Networks, v.3 n.5, p.421-433, Oct. 1997.

[17]. Cheverst, K., Davies, N., Mitchell, K., Friday, "A.: Experiences of developing and deploying a context-aware tourist guide: the GUIDE project, Proceedings of the 6th annual international conference on Mobile computing and networking", August, Boston. (2000)

[18]. Da-Jung Park, Sang-Hee Hwang, Ah-Reum Kim and Byeong-Mo Chang, "A Context-Aware Smart Tourist Guide Application for an Old Palace", Department of Computer Science, Sookmyung Women's University, International Conference on Convergence Information Technology, IEEE 2007.

[19]. Tsvi Kuflik, Julia Sheidin, Sadek Jbara, Dina Goren-Bar, Pnina Soffer, The University of Haifa, Mount Carmel, Haifa, Israel, "Supporting Small Groups in the Museum by Context-Aware Communication Services" Honolulu, Hawaii, USA, IUI'07, January 28–31, 2007.

[20]. Dejene Ejigu, "Context Modeling and Collaborative Context-Aware Services for Pervasive Computing", National Institute of Applied Sciences (INSA de Lyon), 2007.

[21]. Mehdi Jazayeri , "On the Way to Pervasive Computing", Technische Universität Wien

[22]. Paul Chou, Marco Gruteser, Jennifer Lai, Anthony Levas, Scott McFaddin, Claudio Pinhanez, Marisa Viveros, Danny Wong, Sachiko Yoshihama, "BlueSpace: Creating a Personalized and Context-Aware Workspace", Research Division Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo – Zurich

[23]. Jakob E. Bardram, " Applications of Context-Aware Computing in Hospital Work– Examples and Design Principles", Center for Pervasive Healthcare Department of Computer Science, University of Aarhus Aabogade 34, 8200 Aarhus N, Denmark. Proceedings of the 2004 ACM Symposium on Applied Computing

[24]. Jens H. Jahnke, "Towards Context-Aware Computing in Clinical Care", Department of computer Science, University of Victoria, Victoria, B.C. Canada. Position Paper

[25]. Annika Hinze, "Context awareness in Mobile Tourist Information Systems: challenges for User Interaction", George Buchanan, University College London Interaction Centre London, United Kingdom, 2003.

[26]. http://www.archimuse.com/mw2003/workshops/index.html, accessed on September 26, 2008

[27]. http://www.nto.com.et/article.php?, visited on September 28, 2008

[28]. http://www.teleware.com/resources/jargon-buster.asp, accessed on March 12, 2009.

[29]. http://www.parliament.vic.gov.au/SARC/E-Democracy/Final_Report, "Pervasive Computing", accessed on May 04, 2009, Parlament of Victoria.

[30]. W. Schwinger et.al, "Context-awareness in Mobile Tourism Guides – A Comprehensive Survey", Technische Universität Wien, 2001.

[31]. William O. Plymale, "Pervasive computing goes to school", Virginia Tech's Learning Technologies Division, 2005.

[32]. http://www.en.wikipedia.org\Musuem, "Museum", accessed on Sept 02, 2009, Wikimedia Foundation Inc.

[33]. http://www.wordnetweb.princeton.edu/perl/webwn, "Museum", accessed on Sept 02, 2009.

[34]. Institution of Engineering, "Radio Frequency Identification Device Technology (RFID)", the Institution of Electrical Engineers (the IEE), July 2005.

[35]. Motoshi Saeki, Haruhiko Kaiya, Dept of Computer Science Tokyo Institute of Technology, Japan, "On Relationship among Models, Meta-Models, and Ontologies", 2006.

[36]. http://yourdictionary.com/computer/triangulation, accessed on September 29, 2009, "Triangulation", Audio Pronunciations Provided by Love to Know Corp.

[37]. http://wiki.answers.com/Q/List-the-major-advantages-with-layered-approach-to-protocols, "List-the-major-advantage-of-layered-based-approach", accessed on September 30, 2009, Answers Corporation.

[38]. Roger S. Pressman, "Software Engineering: A practitioner's Approach", P. cm- McGraw-Hill series in computer science, 2001.

[39]. Stephen Cranefield et.al, "UML-based Ontology Modeling for Software Agents", Department of Information Science, University of Otago, 2003.

APENDECIES

Appendix A: Questionnaire

1 When does National Museum of Ethiopia established?

____

1.1 Mission ____

1.2 Vision ____

1.3 Core value____

2 What types of heritages are exhibited in the museum? (Religious, political, cultural or other type)____

3 How many rooms are there and how are the heritages arranged or classified? (on shelf and heritage type or others)____

____

____

4 How do you handle whenever new heritage is registered?

____

____

5 Who are major customers of the museum (Foreigners visitors, local visitors, others)____

6 How many employees are there for visitor guiding?________________________

6.1 Background knowledge of heritages

6.1.1 Educational level____

6.1.2 Experience____

6.2 In customer handling

6.2.1 Educational level____

6.2.2 Experience____

6.3 Language skill

6.3.1 Educational level____

6.3.2 Experience____

7 How do you render your service to customer in terms of

7.1 Working time____

7.2 Explanation type for each type of visitor (local foreigner)____

8 How visitors are getting served? (Scenario from a visitor get into museum and finishes)____

____

____

9 How do you compare your visiting service with international museum guide services?

____

____

10 How do you handle conflicts on background knowledge of a particular heritage?

____

____

11 Do you accept comments from visitors? ________________________________

____

12 Is there time limit for visiting session for an individual visitor?_____________

____

13 How do you handle customers' profile?_______________________________

____

14 What do think about the role of heritage background consistency in visiting guide in affecting the perception of visitors? ______________________________

____

15 What mechanism is used to dispatch more explanation to clarifying questions raised by the visitors? ____

____

Appendix B: Ontology of PCMGS


```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
98
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:Her="http://www.owl-ontologies.com/Heritage.owl#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xml:base="http://www.owl-ontologies.com/Heritage.owl">
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://www.w3.org/2003/11/swrl"/>
<owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Created with TopBraid Composer</owl:versionInfo>
</owl:Ontology>
<owl:Class rdf:ID="Location">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="PDA">
<rdfs:subClassOf>
<owl:Class rdf:ID="Device"/>
</rdfs:subClassOf>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Heritage">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Room">
<rdfs:subClassOf rdf:resource="#Location"/>
</owl:Class>
<owl:Class rdf:ID="Answer">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Shelf">
<rdfs:subClassOf rdf:resource="#Location"/>
</owl:Class>
<owl:Class rdf:ID="Heritage_Profile">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Individual_Visitor">
<rdfs:subClassOf>
<owl:Class rdf:ID="Visitor"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Visitor_Profile">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Guide_Service">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:about="#Visitor">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Comment">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Device">
```

```
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Question">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Point">
<rdfs:subClassOf rdf:resource="#Location"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Preference">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="RFID_Tag">
<rdfs:subClassOf rdf:resource="#Device"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Group_Visitor">
<rdfs:subClassOf rdf:resource="#Visitor"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="isDiscoveredFrom">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Preference"/>
<rdfs:range>
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Visitor"/>
<owl:Class rdf:about="#Heritage_Profile"/>
<owl:Class rdf:about="#Visitor_Profile"/>
</owl:intersectionOf>
</owl:Class>
</rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hHasProfile">
<rdfs:range rdf:resource="#Heritage_Profile"/>
<rdfs:domain rdf:resource="#Heritage"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Its profile</rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAskedBy">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Question"/>
<rdfs:range rdf:resource="#Visitor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isLocatedAt">
<rdfs:range>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Location"/>
<owl:Class rdf:about="#Visitor"/>
</owl:unionOf>
</owl:Class>
</rdfs:range>
<rdfs:domain rdf:resource="#Device"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAnsweredAbout">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
```

```
<rdfs:domain rdf:resource="#Answer"/>
<rdfs:range rdf:resource="#Heritage"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAccepted">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:range rdf:resource="#Preference"/>
<rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="owns">
<rdfs:range rdf:resource="#Device"/>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isLocatedInRoom">
<rdfs:domain rdf:resource="#Shelf"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Its location in terms of room number</rdfs:label>
<rdfs:range rdf:resource="#Room"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="containedByShelf">
<rdfs:domain rdf:resource="#Heritage"/>
<rdfs:range rdf:resource="#Shelf"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Its location in terms of shelf number</rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAnswer">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Question"/>
<rdfs:range rdf:resource="#Answer"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAnswerToQuestion">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Answer"/>
<rdfs:range rdf:resource="#Question"/>
<owl:inverseOf rdf:resource="#hasAnswer"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="comments">
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>It refers to the comment spotted by the visitor on a particular heritage</rdfs:label>
<rdfs:range rdf:resource="#Heritage"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="presentedTo">
<rdfs:domain rdf:resource="#Guide_Service"/>
<rdfs:range rdf:resource="#PDA"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasRejected">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="#Preference"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="asks">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<owl:inverseOf rdf:resource="#isAskedBy"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="isPreferenceOf">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Preference"/>
<rdfs:range rdf:resource="#Visitor"/>
<owl:inverseOf>
<owl:ObjectProperty rdf:ID="hasListOfPreference"/>
</owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="trackObjectPoint">
<rdfs:range rdf:resource="#Point"/>
<rdfs:domain rdf:resource="#RFID_Tag"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAnsweredTo">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Answer"/>
<rdfs:range rdf:resource="#Visitor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAssociatedHeritage">
<owl:inverseOf>
<owl:ObjectProperty rdf:ID="hasAssociatedPreference"/>
</owl:inverseOf>
<rdfs:range rdf:resource="#Heritage"/>
<rdfs:domain rdf:resource="#Preference"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isProfileOfHeritage">
<owl:inverseOf rdf:resource="#hHasProfile"/>
<rdfs:range rdf:resource="#Heritage"/>
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasListOfPreference">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="#Preference"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isProfileOfVisitor">
<owl:inverseOf>
<owl:ObjectProperty rdf:ID="vHasProfile"/>
</owl:inverseOf>
<rdfs:range rdf:resource="#Visitor"/>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="vIsLocatedAt">
<rdfs:range rdf:resource="#Location"/>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isLocatedAtRoom">
<rdfs:range rdf:resource="#Room"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
```

```
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ownedBy">
<owl:inverseOf rdf:resource="#owns"/>
<rdfs:range rdf:resource="#Visitor"/>
<rdfs:domain rdf:resource="#Device"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The visitor who owns the device</rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="getService">
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="#Guide_Service"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isStandingInfrontOf">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The shelf at which the visitor stands</rdfs:label>
<rdfs:range rdf:resource="#Shelf"/>
<rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasAssociatedPreference">
<rdfs:range rdf:resource="#Preference"/>
<rdfs:domain rdf:resource="#Heritage"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#vHasProfile">
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="#Visitor_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAskedAbout">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Question"/>
<rdfs:range rdf:resource="#Heritage"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="containsHeritage">
<rdfs:domain rdf:resource="#Shelf"/>
<rdfs:range rdf:resource="#Heritage"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The content of the shelf</rdfs:label>
<owl:inverseOf rdf:resource="#containedByShelf"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isVisitingHeritage">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The heritage which the visitor is visiting</rdfs:label>
<rdfs:range rdf:resource="#Heritage"/>
<rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasVisited">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="#Heritage"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isPreparedFrom">
<rdfs:range>
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Heritage_Profile"/>
<owl:Class rdf:about="#Visitor_Profile"/>
```

```
</owl:intersectionOf>
</owl:Class>
</rdfs:range>
<rdfs:domain rdf:resource="#Guide_Service"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="timeToSpentHere">
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#time"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The time to which the visititor wants to stay in the musuem</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="deviceId">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="objectFunction">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Functionality of the object</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="institutionName">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Name of the body the object donated by</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="isNewComer">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>It refers to the situation if the visitor is new or not</rdfs:label>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="shortDescription">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>A short description given for visitors</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="accessNumber">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The access number by which an object can be accessed</rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="rYCordinate">
<rdfs:domain rdf:resource="#Room"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Y cordinate of a room</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="shelfNumber">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Shelf number</rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:domain rdf:resource="#Shelf"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="material">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>The material the object made of</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="rHeight">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Room Height</rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:domain rdf:resource="#Room"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="vId">
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>visitor identification number</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="visitingSessionStartTime">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="roomNumber">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Room Number</rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:domain rdf:resource="#Room"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sXCordinate">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:domain rdf:resource="#Shelf"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="y">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:domain rdf:resource="#Point"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sWidth">
<rdfs:domain rdf:resource="#Shelf"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>shelf width</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tempLocation">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Name of current location, if the object is moved away</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="age">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tagNumber">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasNegativeFactor">
```

```
&lt;rdfs:domain rdf:resource="#Visitor_Profile"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="hName"&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Heritage Name&lt;/rdfs:label&gt;
&lt;rdfs:domain rdf:resource="#Heritage"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="educationalLevel"&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The educational background of the visitor&lt;/rdfs:label&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
&lt;rdfs:domain rdf:resource="#Visitor_Profile"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="nativeCountry"&gt;
&lt;rdfs:domain rdf:resource="#Visitor_Profile"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The native country of the visitor&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="languagePreference"&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:label&gt;
&lt;rdfs:domain rdf:resource="#Visitor_Profile"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="heritageCommentValue"&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:label&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/&gt;
&lt;rdfs:domain rdf:resource="#Heritage"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="sYCordinate"&gt;
&lt;rdfs:domain rdf:resource="#Shelf"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>X cordinate of a Shelf&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="heritagePreference"&gt;
&lt;rdfs:domain rdf:resource="#Visitor_Profile"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The heritage type preference of the visitor&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="pId"&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:label&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
&lt;rdfs:domain rdf:resource="#Preference"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="acquisitionSource"&gt;
&lt;rdfs:domain rdf:resource="#Heritage_Profile"/&gt;
&lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The source entity the objec acquired&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="questionType"&gt;
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>&lt;/rdfs:label&gt;
&lt;rdfs:domain rdf:resource="#Question"/&gt;
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="image">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="x">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:domain rdf:resource="#Point"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="aquisitionDate">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Acquisition Date</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="shortPrefDescription">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Preference"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="rWidth">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Room width</rdfs:label>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:domain rdf:resource="#Room"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="lastName">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Visitor's last name</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="distingushingFeature">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The feature to which the object is distinguished</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="objectTechnique">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Technique of development</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="signature">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Signature of Donors</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="alreadyVisitedMusuem">
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Already visited musuems by the visitor</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tempLocNotice">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Notice over the reason of moving object to the temp location</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="aquisitionMethod">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Method of Object Acquisition</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="confidenceLevel">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Preference"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="estimatedVSpeed">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="finishes">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="rXCordinate">
<rdfs:domain rdf:resource="#Room"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>X cordinate of a room</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="serviceId">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Guide_Service"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="permanentPlace">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Permanent place</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hCaption">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Heritage Caption</rdfs:label>
<rdfs:domain rdf:resource="#Heritage_Profile"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="experienceOrSkill">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Visitor_Profile"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sHeight">
<rdfs:domain rdf:resource="#Shelf"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Shelf height</rdfs:label>
</owl:DatatypeProperty>
```

```xml
<owl:DatatypeProperty rdf:ID="aquisitionPlace">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Acquisition place</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="measurement">
<rdfs:domain rdf:resource="#Heritage_Profile"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Measuring scale (weight, height,....)</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hId">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Heritage Identification Number</rdfs:label>
<rdfs:domain rdf:resource="#Heritage"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="firstName">
<rdfs:domain rdf:resource="#Visitor"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>First name of the visitor</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="visitingSpeed">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<rdfs:domain rdf:resource="#Visitor"/>
</owl:DatatypeProperty>
<Her:Preference rdf:ID="Preference_1">
<Her:shortPrefDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>It is first preference of visitors who has archiological heritage type
preference</Her:shortPrefDescription>
<Her:pId rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>p0001</Her:pId>
<Her:isPreferenceOf>
<Her:Visitor rdf:ID="VisitorInstance1">
<Her:owns>
<Her:RFID_Tag rdf:ID="RFIDTagInstance1">
<Her:isLocatedAt>
<Her:Shelf rdf:ID="shelfInstance1">
<Her:sHeight rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>4.0</Her:sHeight>
<Her:sWidth rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>4.0</Her:sWidth>
<Her:sXCordinate rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>0.0</Her:sXCordinate>
<Her:sYCordinate rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>0.0</Her:sYCordinate>
<Her:shelfNumber rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</Her:shelfNumber>
<Her:isLocatedInRoom>
<Her:Room rdf:ID="roomInstance1">
<Her:rHeight rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>8.0</Her:rHeight>
<Her:rWidth rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>4.0</Her:rWidth>
<Her:rXCordinate rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>0.0</Her:rXCordinate>
<Her:rYCordinate rdf:datatype="http://www.w3.org/2001/XMLSchema#double"
>0.0</Her:rYCordinate>
<Her:roomNumber rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
```

>1</Her:roomNumber>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:Room>
</Her:isLocatedInRoom>
<Her:containsHeritage>
<Her:Heritage rdf:ID="heritageInstance1">
<Her:heritageCommentValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</Her:heritageCommentValue>
<Her:hId rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>h0001</Her:hId>
<Her:hName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Lucy</Her:hName>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:Heritage>
</Her:containsHeritage>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:Shelf>
</Her:isLocatedAt>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:RFID_Tag>
</Her:owns>
<Her:visitingSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>20</Her:visitingSpeed>
<Her:isVisitingHeritage rdf:resource="#heritageInstance1"/>
<Her:finishes rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>false</Her:finishes>
<Her:lastName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>William</Her:lastName>
110
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<Her:firstName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>David</Her:firstName>
</Her:Visitor>
</Her:isPreferenceOf>
<Her:hasAssociatedHeritage rdf:resource="#heritageInstance1"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:Preference>
<Her:Visitor_Profile rdf:ID="visitorProfileInstance1">
<Her:timeToSpentHere rdf:datatype="http://www.w3.org/2001/XMLSchema#time"
>11:59:19.639</Her:timeToSpentHere>
<Her:alreadyVisitedMusuem rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Not at all</Her:alreadyVisitedMusuem>
<Her:educationalLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Professor</Her:educationalLevel>
<Her:vId rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>v0001</Her:vId>
<Her:isNewComer rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>true</Her:isNewComer>
<Her:tagNumber rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>RFID0001</Her:tagNumber>
<Her:heritagePreference rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Archeological Heritages</Her:heritagePreference>
<Her:isProfileOfVisitor rdf:resource="#VisitorInstance1"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<Her:hasNegativeFactor rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>3</Her:hasNegativeFactor>

```
<Her:nativeCountry rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Canada</Her:nativeCountry>
<Her:age rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>32</Her:age>
<Her:visitingSessionStartTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2009-06-25T11:59:57.926</Her:visitingSessionStartTime>
<Her:languagePreference rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>English</Her:languagePreference>
</Her:Visitor_Profile>
<Her:PDA rdf:ID="PDAInstance1">
<Her:isLocatedAt rdf:resource="#shelfInstance1"/>
<Her:ownedBy rdf:resource="#VisitorInstance1"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</Her:PDA>
<Her:Heritage_Profile rdf:ID="heritageProfileInstance1">
<Her:distingushingFeature rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The first person....</Her:distingushingFeature>
<Her:shortDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Named as Lucy in English and Dinknesh in native language Amharic meaning "You are wonderful
!". Its weigh is 52 kg and height is 1.79 m. It is found 4.69 million years before around Afar region in
Ethiopia.</Her:shortDescription>
<Her:measurement rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>1.79 m height and 52 kg weight</Her:measurement>
<Her:hCaption rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Lucy /Dinknesh</Her:hCaption>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
<Her:signature rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>registered</Her:signature>
<Her:objectTechnique rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Natural</Her:objectTechnique>
<Her:estimatedVSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>30</Her:estimatedVSpeed>
<Her:aquisitionDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
>2008-08-25</Her:aquisitionDate>
<Her:aquisitionPlace rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Afar Hadar</Her:aquisitionPlace>
<Her:isProfileOfHeritage rdf:resource="#heritageInstance1"/>
<Her:objectFunction rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Skeleton</Her:objectFunction>
<Her:permanentPlace rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Ethiopia</Her:permanentPlace>
</Her:Heritage_Profile>
<rdf:Description rdf:ID="deleteThis">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></rdfs:label>
</rdf:Description>
</rdf:RDF>
<!-- Created with TopBraid Composer -->
```

Appendix C: List of Java Classes Implemented

Class Name	Description
Heritage. java	Represents heritage attributes important to abstract museum heritage
InstanceBuilder.java	A class that holds methods to create instances of classes in PCMGS environment
Preference. java	Represents preference attributes important to abstract visitor preference
PreferenceAnalyzer.java	A class that contains important methods to analyze Confidence Level of each preference such as analyzer, possibilityManager, calculateCL, CalculateCM…
PreferenceDiscoverer.java	A class that contains important methods to discover possible preferences of a visitor from previous visitors' profile. It uses WEKA API libraries
RFIDReader-Simulator.java	A class that simulates the functionalities of RFID readers. It continuously generates random numbers to represent tag identification number and its corresponding X-Y coordinate
Room. java	Represents rooms found at the museum. It is designed to hold height and width of a room.
Shelf. java	Represents shelves found at the museum. It is designed to hold height and width of a shelf.
Service. java	A class that contains important methods such as service provider, service coordinator,…
VIDPool.java	A class that contain

	important method such as VID pool handler to handle the visitor identification number.
Visitor. java	An abstract representation of a person who is visiting at the museum.

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: ____

Signature: ____

Date: ____

Confirmed by Advisor:

Name: ____

Signature: ____

Date: ____

Place and date of submission: Addis Ababa, December 2009.