CONTEXT-AWARE MOBILE LEARNING ON THE SEMANTIC WEB

By

Xiaoyun Zhang

A Thesis Submitted to the Department of Electrical Engineering

in Partial Fulfillment of the Requirement for the Degree of

Master of Science

at

Lakehead University

Thunder Bay, Ontario, Canada

April 2008

Your file Votre référence
ISBN: 978-0-494-43414-7
Our file Notre référence
ISBN: 978-0-494-43414-7

ACKNOWLEDGMENTS:

This work would not have been possible without the aid of my supervisor, Dr. R. Benlamri; I am thankful for his insight, guidance and support.

Abstract

Progress made in Semantic Web technologies and Ubiquitous Computing has lead to the development of mobile learning services that can adapt to the learner's background, learner's needs, and surrounding environment. In particular, the emerging techniques from these two technologies have the potential to revolutionize the way mobile learning services available on the web are discovered, adapted, and delivered according to context. Context acquisition and management, conceptual knowledge modeling and reasoning, and adaptive services discovery are the main ingredients for designing such context-aware mobile learning systems. However, a number of challenges are still facing the research community in this field. These can be summarized in the following: (i) current mobile learning services act as passive components rather than active components that can be embedded with context awareness mechanisms, (ii) existing approaches for service composition neglect contextual information on surrounding environment, and (iii) lack of context modeling and reasoning techniques for integrating the various contextual features for better personalization. In this thesis an attempt is made to solve the above-mentioned problems. These challenges are addressed by proposing a personalized mobile learning system based on a global ontology space to aggregate and manage context information related to the learner, the used device, the surrounding environment, and the task at hand. The system adopts a unified reasoning mechanism, around the global ontology space, in order to adapt the learning sequence and the learning content based on the learner profile and the perceived contextual information. The adopted approach for ontology reasoning aims at achieving two types of adaptations – system-centric adaptation and – learner-centric adaptation. These are implemented on a Run-Time Environment that identifies new contextual changes and translates them into new adaptation constraints. We developed and tested our system on a number of subject-domain ontologies using various learning scenarios, and the obtained experimental results are very promising.

Keywords: Semantic Web, Ontology, Mobile Learning, Ubiquitous Computing, Context Modeling and Management, Ontology-based Reasoning, Web Services.

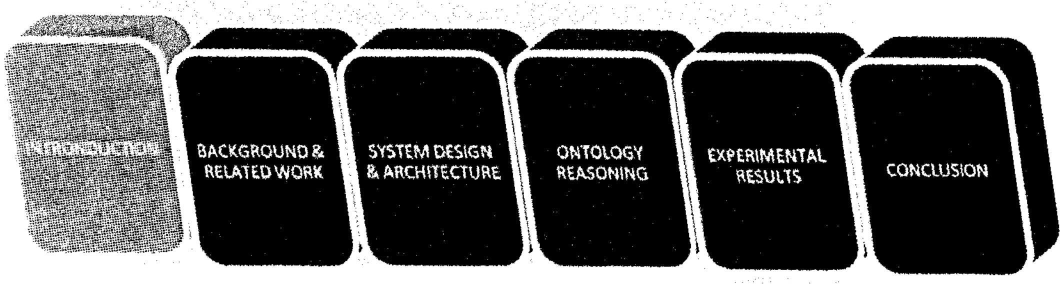
TABLE OF CONTENTS:

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION



1.1 Motivation and Objectives

This study is concerned with the design and implementation of a mobile learning system tailored to the needs and context of learners. Context acquisition and management [1-3], conceptual knowledge modeling [4-5] for personalized learning, and adaptive information discovery are the most important requirements for designing such an intelligent mobile learning system. The goal of the research community in the field of mobile learning is to develop ubiquitous learning environments capable of providing useful learning resources on demand, anywhere, in a learner-driven context and on a learner's schedule [1]. In the ubiquitous environment, the context-awareness framework needs to aggregate and integrate context information related to the learner (i.e. preferred language, previously conducted learning interactions), the used device (i.e. operating system, current available memory size), the surrounding environment (i.e. varying network bandwidth, location), and the task at hand (i.e. current learner's interaction, goal). Research work in this field has been dominated by the use of ontologies and other related semantic web technologies for context-awareness. Many approaches to context modeling have been considered [4-7].

Another important aspect of mobile learning is the design and deployment of mobile web services. This field is becoming a very active area of research and development [6-8]. However, some challenging aspects are facing the research community in the area of personalized mobile services. These are:

- Current mobile web services act as passive components rather than active components that can be embedded with context awareness mechanisms.

- Existing approaches for service composition typically facilitate choreography only, while neglecting contextual information on users, environment, and services.

- Lack of context modeling techniques and middleware for integrating the various contextual features for better personalization.

This study proposes a solution to the above mentioned problems by developing a personalized mobile learning system with semantic-rich awareness information. In particular, this study focuses on a new context modeling and ontology-based reasoning mechanism. This approach is based on the fact that context is not simply the state of a predefined environment with a fixed set of interaction resources, but it is part of a process that is interacting with an ever-changing environment composed of a set of heterogeneous atomic context elements [9]. Therefore, the proposed personalized mobile learning system is based on a Run-Time Environment (RTE) that identifies the new contextual features and translates them in to new adaptation constraints. For instance, the system automatically updates the perceived – device or – environment context elements and uses them to re-adjust inferred metadata that adapts the search for those compatible learning resources. The main contributions of our work are as follows:

- A unified ontology space for context integration and aggregation based on learner context, activity context, device context, and environment context. In addition, this ontology space contains a domain ontology used to define the subject domain area of interest.

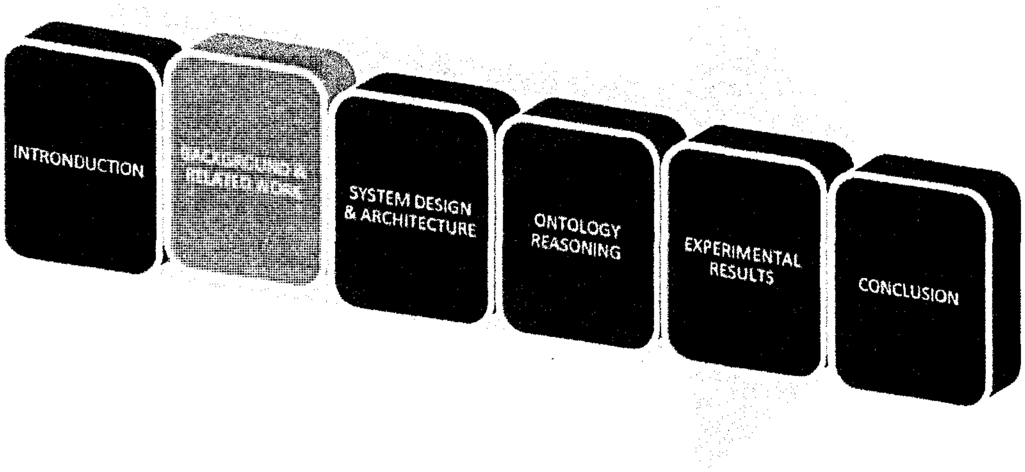
- Efficient learning sequence management and subject domain knowledge representation at different granularity levels to suit learners with various cognitive skills.
- A Run-Time Environment model for context management to permit context perception, context identification, and context adaptation. The system uses ontology reasoning to infer high level context at the semantic level to achieve both system-centric adaptation and learner-centric adaptation.

1.2 Structure of the Thesis

Chapter 2 introduces some basic background for Semantic Web technologies such as the Resource Description Framework (RDF), Web Ontology Language (OWL), and Semantic Web Rule Language (SWRL). It also describes current research work in the field of mobile learning. Chapter 3 presents the overall system architecture and the main functions of the system. It describes the approach used for context acquisition and modeling. It also describes a global ontology space for context integration and aggregation based on user context, activity context, device context, and environment context. Chapter 4 describes the reasoning and learning strategies used to personalize learning. Chapter 5 includes a number of case studies that illustrate the main functions of the system. Chapter 5 also provides a performance evaluation study by comparing our system with some other existing systems. Finally, conclusions from the work are drawn and further research work is suggested.

CHAPTER 2

BACKGROUND & RELATED WORK



This chapter introduces basic technical background in the area of semantic web. It then overviews the related research work in the field, and finally, it introduces the approach adopted in this study for the design and implementation of a context-aware mobile learning system.

2.1 Semantic Web

The semantic web is an extension of the World Wide Web, in which information and services are given well-defined meaning, making it possible for the web to understand the requests of people, and enabling computer and people to work in cooperation [10-11]. To make possible the creation of the semantic web, the World Wide Web Consortium (W3C) has been actively working on the definition of open standards [12]. Based on these standards, the semantic web will empower intelligent services such as search agents, information filters, and knowledge management systems.

2.2 Current Web v.s. Semantic Web

Today's web is primarily composed of documents written in presentation mark-up languages like Hyper Text Mark-up Language (HTML). HTML was designed for human interpretation and use. Each web page has a Uniform Resource Locator (URL) address and can be easily accessed by people. Humans can read information from web pages, understand them and process item, but the machine is not smart enough to handle the above task. It can not read, analyze, and interpret the meaning of the information in presentation mark-up language form. Semantic web is used to express resources in a machine-processable format that can be used by computers not only for display purposes, but also for interoperability and integration between systems and applications [12]. It will bring a structure to the meaningful web resources, and sets the inference rules for automatic reasoning. Table 2.1 describes the overall difference between the current web and the semantic web.

Current Web	**Semantic Web**
Context Sharing	Resource Sharing
Presentation Mark-UP	Semantic Mark-UP
Human Interaction	Machine Processable
Product / Download	Dynamic Web-Services
Producer / Consumer	Collaborative virtual Communities
Bolt on security	Detailed Security Model

Table 2.1 Current Web v.s. Semantic Web

2.3 Semantic Web Technologies

The main web technologies used by the semantic web are Uniform Resource Identifier (URI), Extensible Makeup Language (XML), RDF and OWL. People, places, and things in the physical world will have online representations identified by URIs. XML is a mark-up language that provides syntax for content structure within documents. It allows everyone using open standard syntax to create their own documents. RDF is a data model for representing resource's information on the web [13]. Many Resources need to be processed by applications instead of only displayed to people. The motivation of RDF is to create a format for making assertions about resources and to combine data from several applications. Subject, Predicate and Object are three main entities for representing an RDF statement. The subject and predicate are URIs, while the object can be a URI or a literal. In RDF, there are only two types of data for property values: string and URIs. Figure 2.1 shows a cell phone with a resolution of 320 * 240 pixels.

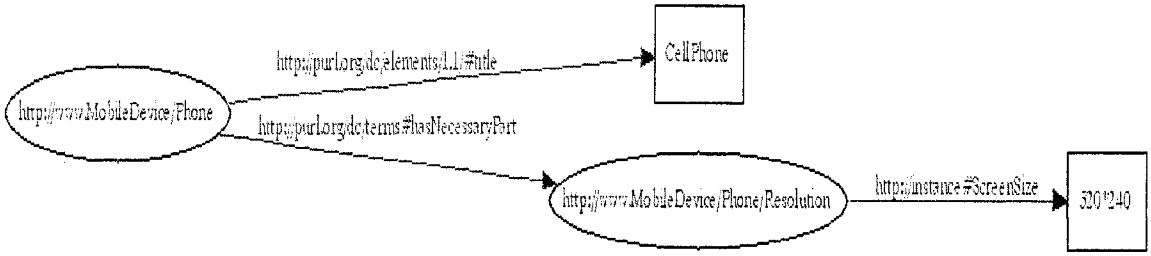

http://purl.org/dc/elements/1.1/#title
CellPhone
http://www.MobileDevice/Phone
http://purl.org/dc/terms#hasNecessaryPart
http://www.MobileDevice/Phone/Resolution
http://instance#ScreenSize
320*240

Figure 2.1 Figure 2.1 RDF Graph for a Mobile Phone

The Semantic Web will build on XML's ability to define customized tagging schemes and an RDF's flexible approach to representing data. The first level above RDF required for semantic web is an ontology language which can formally describe the meaning of terminology used in the web document [14]. OWL can be used to explicitly describe an ontology that is a representation of concepts and their relationships. OWL provides three increasingly expressive sublanguages that are OWL Lite, Web Ontology Language Description Language (OWL DL), OWL Full. The ontology developer needs to consider the sublanguages that best suit their need. The OWL Lite is a basic sublanguage that can support a classification hierarchy and simple features. The OWL DL has more advantages for maximizing expressiveness without losing computational completeness and decidability of reasoning systems. The OWL Full developed for users who want maximum expressiveness and syntactic freedom of RDF with no computational guarantees. Figure 2.2 represents the seven layers structure for the semantic web.

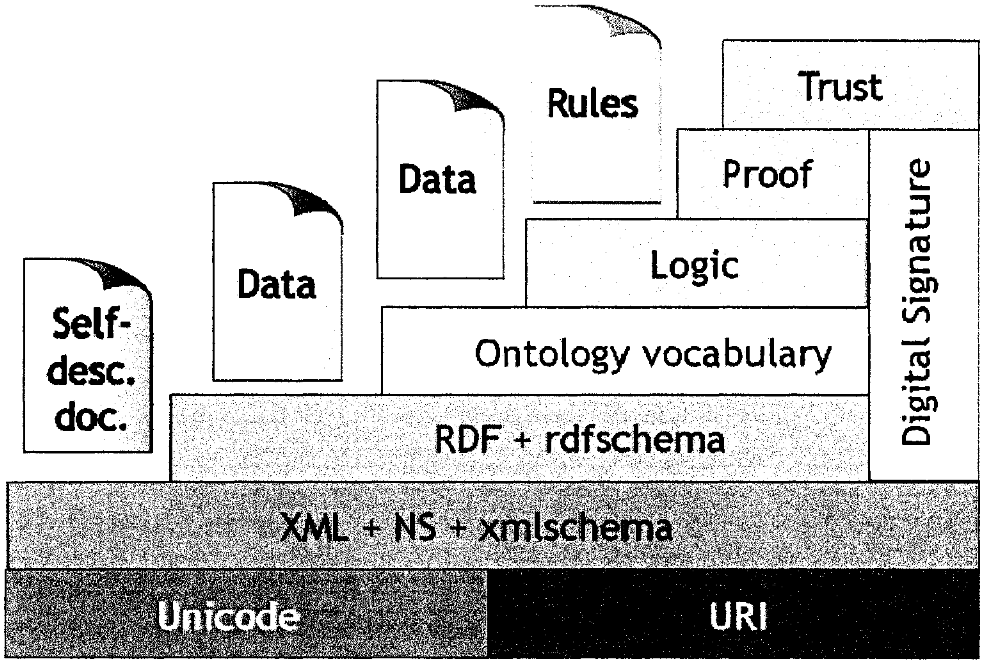


Figure 2.2 Semantic Web Structure

2.4 Reasoning Techniques

Ontology reasoning approaches can use various kinds of logic to support inference; description logic, first order logic, temporal logic, and spatial logic to name a few [15]. Although there are many ontology reasoning languages [16-19], such as SWRL, Rule Markup Language (RuleML), and Description Logic Programs (DLP), SWRL has been proposed as the basic rules language for the semantic web. It is a combination of the OWL DL and OWL Lite sublanguages of OWL with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. While DLP is the intersection of Horn logic [17-18] and OWL, SWRL is (roughly) the union of them. In DLP, the resultant language is a very peculiar looking description logic and rather inexpressive language overall. It is hard to see the restrictions are either natural or satisfying. Contrariwise, SWRL retains the full power of OWL DL, but at the price of decidability and practical implementations [16]. SWRL is a logic language with rules expressed in the following format:

$$A_1,...A_n \rightarrow B$$

Where $A_1,...A_n$ and B are atomic formulas. The set $\{A_1,...A_n\}$ is referred to as the antecedent (body) of a rule, and B is a consequent (head) of a rule. The atoms $A_1,...A_n$ and B can be of the form C(x), P(x,y), sameAs(x,y), differentFrom(x,y), or builtIn(r,x,...), where C is an OWL description or data range, P is an OWL property, r is a built-in relation, x and y are either variables, OWL class individuals or data values as appropriate [19]. The following is an instance described by a SWRL rule. In this SWRL rule, the concept person has been captured using an OWL class called *Person*; the *parent, sister* and *aunt* relationships can be expressed using OWL object properties *hasParent, hasSister,* and *hasAunt* respectively.

```
Rule-1:
Person(?x)^hasParent(?x,?y)^hasSister(?y,?z)→hasAunt(?x,?z)
```

2.5 Related Work

Semantic web technologies have been used in recent years to develop personalized learning systems. In particular, ontology-based approaches have been used for context modeling and management; and logic approaches have been used for ontology inference and reasoning [20-25].

Such applications which take advantage of the context are called context-dependent or context-aware applications and lead us to the development of context-aware systems [26-29]. In Shehzad and Ngo, it is advocated that the use of formal modeling in context aware systems will bring many advantages to the area [28], mainly when focusing on solving the following problems: (1) sharing of common information semantics; (2) testability of formalized knowledge; and (3) emergence of a pool of consistent contextual knowledge available to different context-aware systems. Their work discussed context models for the home domain and shows how it entails implicit reasoning. The formalized context model is based on categorized context entities such as agent, devices, environment, location and time. In particular, they defined the contextual information hierarchy among sensor based information, elementary context, and composite context. In a similar study, four types of context-awareness models are identified by Lee et al. for ubiquitous environments: (1) the basic type is Sense-Context that is gathered from the sensors; (2) Combined-Context that is calculated by Sensed-Context and the representation of calculation formula is represented in SWRL; (3) Inferred-Context that is inferred by Sensed-Context and the representation of inference is also represented in SWRL; and (4) Learned-Context that is made by a learning algorithm such as Decision-Tree (DT) or Neural-Network (NN) [29].

Yang proposed a work for context model and context acquisition mechanism for collecting contextual information at run time [30]. In particular, their work does not only provide an ontology based context model but also utilizes two context acquisition methods context detection and context extraction, for obtaining various contextual information [30]. They developed two types of context ontologies: learner ontology, and service ontology. The learner ontology consists of learner profile, preferences, Quality of

Learning Services (QoLS), environment, and services. The service ontology consists of service profiles, and QoLS. In their work, the context detection is tracked from two sides: server side (i.e. analysis of previous work), and client side (i.e. sense learners' surrounding environment). The context is extracted from the learner's default context based on the preferences and derived contextual information from the calendar profile, and social profile.

In the field of personalized mobile learning [31-34], Yu and Nakamura developed a personalized and complete learning system to support mobile learners [31]. The system consists mainly of three ontologies (Learner ontology, Learning Content Ontology, and Domain Ontology), five rules for semantic relevance calculation, and an algorithm for generating the learning path. For ontology modeling, they designed a learner ontology that depicts context about the learner (i.e. subject or particular content already mastered, learning goals, available learning time, current location, desired learning style, and learning interests). For the learning content ontology, they defined a relation *hasPrerequisite* that describes context dependency information. The domain ontology is proposed to integrate existing consensus domain ontologies such as computer science and chemistry. For semantic relevance calculation, their work adopts the following steps: (1) map the user's goal to the domain ontology; (2) locate the subject of the learning content in the domain ontology; (3) estimate the conceptual proximity between the mapped element and the subject node of the learning content [31]. According to their algorithm, the system can generate a learning path connecting with prerequisite contexts (*hasPrerequisite* relations) and the target learning context. In another study by Henze and Dolog, the proposed system uses three types of ontologies (domain, user, and observation) to realize dynamically personalized e-learning tasks on the semantic web [28]. They suggested a framework for such adaptive or personalized educational hypermedia system based on a number of semantic web techniques. In particular, they show how rules can be enabled to reason over distributed information resources in order to dynamically derive hypertext relations which are used to recommend a sequence of learning tasks.

Berri and Benlamri have developed a learning system where extracted conceptual knowledge from a source ontology is efficiently used by firing a set of rules based on the learner profile to recommend a learning path [35]. In an extension of the same work, Basaeed et al. have divided the learner context into two models: learning model (i.e. authentication information, age) and learning preferences (i.e. learning style, difficulty level). The device context is described in the terms of its device type and its capabilities (i.e. navigation tools, bandwidth limitations). The system uses learner ontology, device ontology, and domain ontology to enable better learner modeling, efficient context acquisition and management, and reusable customized learning content [1]. The main component in their system is the learning web constructor that operates in three-steps: context sensing, context reasoning, and context adaptation. The system matches the learner's goal to the concepts in the ontology based on three relationships: necessary part-of, part-of, and is-a. Then, the importance of Learning Objects (LOs) is inferred by using rules that are retrieved from "context reasoning". Finally, the system considers the time issue, and then generates a learning path using necessary part-of, part-of, and is-a contexts (assigned from higher to lower importance levels respectively). Accordingly, their system achieves the initial goal providing "just enough, just in time, just for me" learning delivery [1].

To achieve true context awareness, however, mobile systems must produce reliable information in the presence of uncertainty, rapidly changing, and environment partially true data from several sources [36-39]. In Korpipaa et al., fuzzy sets or crisp limits are introduced for quantizing extracted features [36], where resource servers use one of two methods for quantization: set crisp limits (true-false), or apply a fuzzy set for features (a truth value between 0 and 1). The resource servers use an unstructured raw measurement data, and return a representation defined in the context ontology for context management. In a similar study by Pan and Stoilos, a system called f-SWRL that is a fuzzy extension to SWRL is developed [37]. The system includes fuzzy assertions and fuzzy rules. In their work, the atoms in f-SWRL can include a "weight" that is a truth value between 0 and 1. The "weight" represents the "importance" of the atom in a rule. For example:

$$Rich(?\,p) * 0.5 \wedge Healthy(?\,p) * 0.9 \rightarrow Happy(?\,p)$$

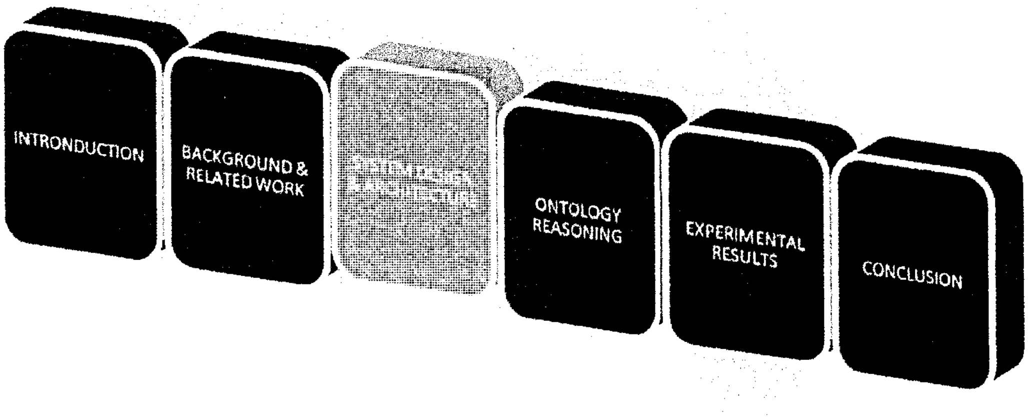
where values 0.5 and 0.9 represents the weights for atoms $Rich(?\,p)$ and $Healthy(?\,p)$.

To design and implement personalized mobile learning systems on the semantic web, there are at least three related research areas which need to be considered. These are: context-awareness frameworks for ubiquitous environments, adaptive information discovery, and ontology-based reasoning mechanisms. It should also be noted that the characteristics and requirements of mobile learning are different and far more complex than those of traditional learning systems. For example, low bandwidth, limited screen resolution, and unsecured wireless communication are a just few technological constraints that make up the system's complexity. The proposed system attempts to solve some of the above mentioned challenges.

Our approach uses three hierarchical levels – atomic level, composite level, and high level. Sensor based information, elementary context, and composite context are defined for each contextual information hierarchy. In addition, the system integrate knowledge related to the learner, learning activity, used mobile device, and surrounding environment and it defines them at the semantic level using a global interrelated ontology space. Our system uses ontology reasoning to infer high level context at semantic level for both system-centric adaptations and learner-centric adaptations. System-centric adaptations are used to ensure searched learning resources are suitable for the system-centric metadata. For learner-centric adaptations, the system uses learner's tacit knowledge to build a learning path for better personalization.

CHAPTER 3

SYSTEM DESIGN & ARCHITECTURE



This chapter presents the overall system architecture and provides a detailed description of its main components. It first describes the context sensing and acquisition system. It then presents the major software components of the system design, and finally, it introduces the user interaction with the system. This chapter also describes the approach used for context acquisition and modeling. It designed a global ontology space based on learner context, activity context, device context, environment context, and domain context.

3.1 Overall System Architecture

This section overviews the main components of the system architecture. Figure 3.1 describes the proposed learning system which consists of the context sensing and acquisition unit, the ontology reasoning unit, and the service/resource discovery and adaptation unit.

The context sensing and acquisition unit consists of three hierarchical levels – atomic level, composite level, and high level. Atomic level retrieves the user interaction and senses atomic context elements from different sources. At the composite level, the system uses inference, computation, and a learning technique to translate atomic context elements into meaningful symbolic context information. High level context consists of four-tuple $C_t = (C_L, C_D, C_E, C_A)_t$ which are built out of configurations of composite context elements sensed at time t around a specific learning domain [40]. It should be noted that C_L is learner context, C_D is device context, C_E is environment context, and C_A is activity context. The above hierarchy is denoted by the context sensing and acquisition units shown in Figure 3.1.

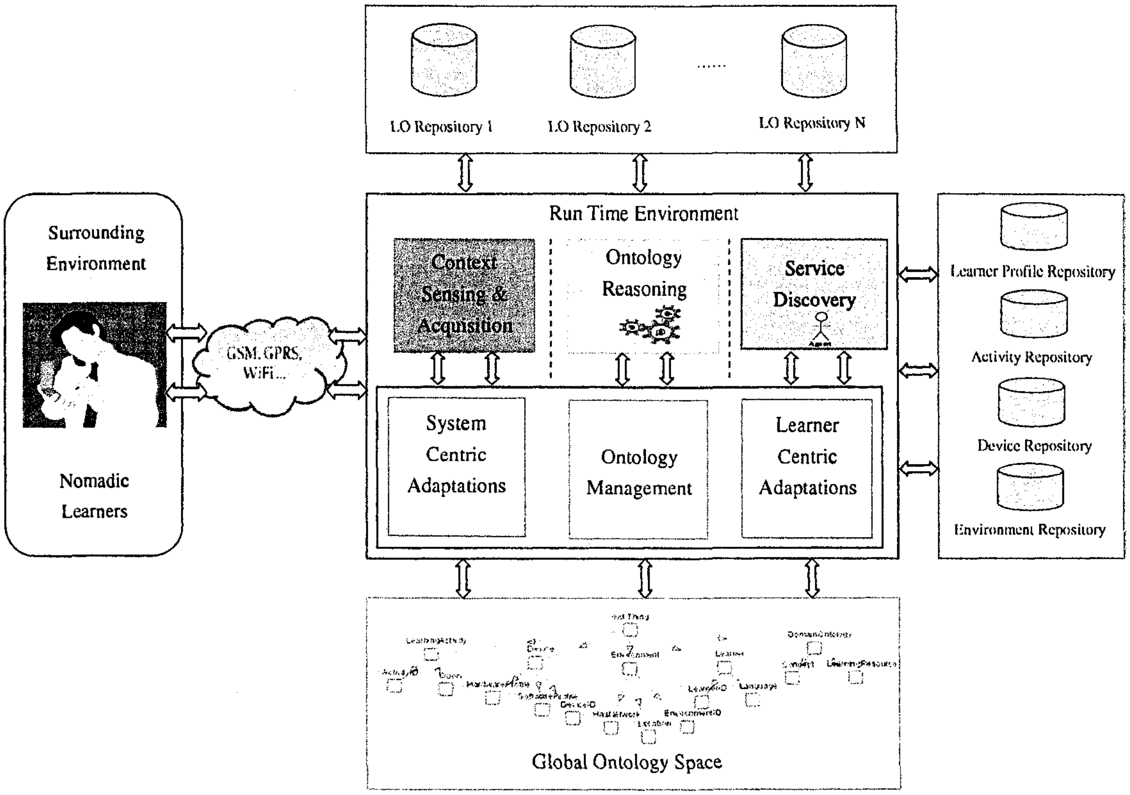


Figure 3.1 System Architecture

The system uses a global ontology space to enable efficient context modeling and management. It also adopts a unified reasoning mechanism to share and reuse personalized learning content. As shown in Figure 3.1, the global ontology space consists of a domain ontology and four interrelated sub-ontologies that are learner ontology, device ontology, environment ontology and activity ontology. In this study, the system used the OWL DL for describing the global ontology space. Ontology-based reasoning is a key design for our global ontology space to enable personalized learning that can be achieved in two different aspects: – system-centric adaptation – and learner-centric adaptation. In system-centric adaptations, the system ensures searched learning resources are suitable for the system-centric metadata generated from perceived device and environment atomic context elements. For the learner-centric adaptations, the system uses learner's tacit knowledge to build a learning path for better personalization. This is achieved in terms of a sequence of service discovery and adaptations as described in Figure 3.1.

3.1.1 Operational environment

This section describes the main software components for setting up our system environment. Figure 3.2 describes the major technologies used to build our system as well as the basic system processing steps.

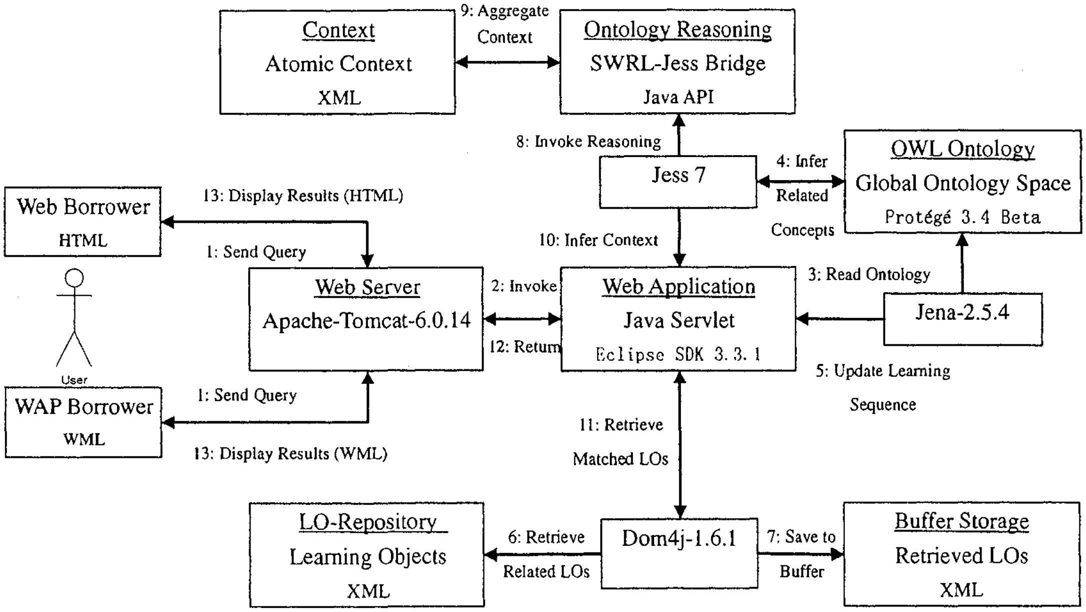


Figure 3.2 Software Components

The system uses Apache – Tomcat 6.0.14 as the servlet container which can be manually started and stopped. Java is the main programming language in our system. System opted for an IDE (Eclipse SDK 3.3.1) as software development platform. For context modeling, system uses Protégé 3.4 Beta for ontology editing and knowledge acquisition purposes. Jena API is a Java framework for building semantic web application. It is used to read the global ontology space and to create prerequisite individuals. Dom4j is used to process XML files such as reading data from learning resources and writing retrieved data to buffer storage. For ontology reasoning, the system has utilized SWRL Tab of Protégé to build the SWRL rules. Jess is a rule engine used as an interactive tool for manipulating Protégé ontologies. SWRL – Jess Bridge is a subcomponent of the SWRL Tab that provides a bridge between an OWL model with SWRL rules and rule engine Jess 7.0. The sequence of processing between these main components as shown in Figure 3.2 is described in more details in the next chapter.

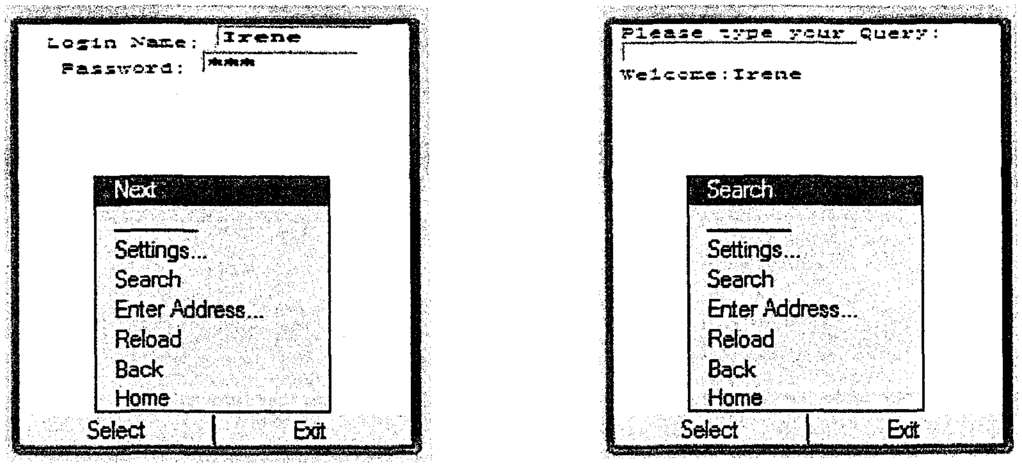
3.1.2 User interface design

This section describes the user interaction of our system. The user interface uses a menu-driven interface to control the main functions of the system and to help the user navigate in the learning web. Figure 3.3 and Figure 3.4 show the major functions which consist of:

- Next: verify user's password and turn into main page when user submit password correctly.
- Search: Submit user typed query to web server.
- Go with Recommend: Submit user selected concept to web server.
- Default: Reset the user's profile to initial state.

- Back: Return to previous page.

The learner is exposed to an interface like the one shown in Figure 3.3.a when the user opens the WAP 2.0 (Wireless Application Protocol) browser using his handheld device. Figure 3.3.b shows the welcome page after authentication is passed. The user can then type any keyword in the input query-part and submit it to the server by using the *Search* button. Figure 3.4.a shows the related search results after the server has received the user's query. The system offers many optional concept-keyword and related learning resources. For the optional concepts, the user can select *Go with Recommendation* button to acquire more learning resources. The related learning resources represent all learning resources about the user's query. Any consumed learning resource or concept will not be displayed to the user in this part. The *Default* button can help to reset the user's profile to initial state. This button deletes all history about consumed learning resources. Figure 3.4.b selects a specific learning resource when user clicks learning resource's title with URI link. The user can be returned to the previous page when selecting the *Back* button.



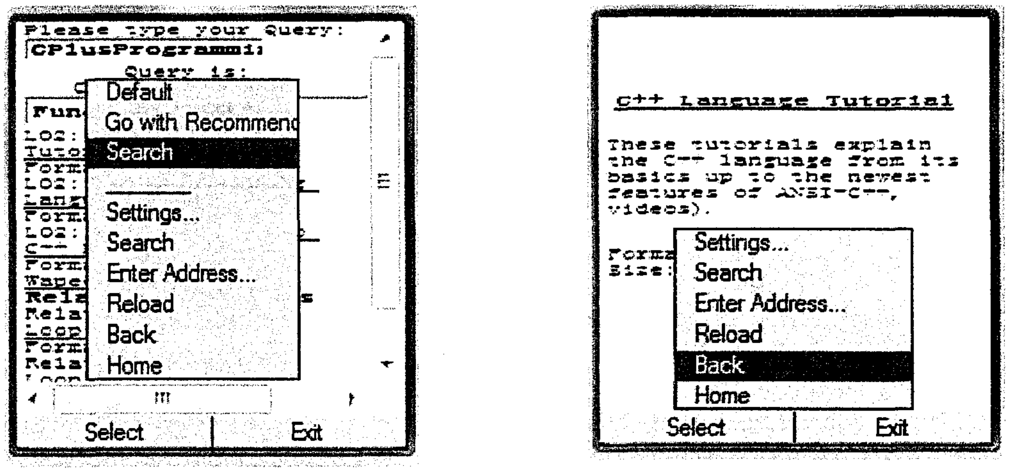
Login Name: Irene
Password: ***

Please type your Query:
Welcome:Irene

(a)

(b)

Figure3.3 User Interface (1)


Please type your Query:
CPlusProgrammi

Query is:

C++ Language Tutorial

These tutorials explain the C++ language from its basics up to the newest features of ANSI-C++, videos).

(a)

(b)

Figure3.4 User Interface (2)

3.2 Context Acquisition & Representation

According to Schilit, context is referred to as any information that can be used to characterize the situation of any entity where an entity can be a person, place, and physical or computational object [30]. While context entities are conceptual entities, the information provided by them is called contextual information [28]. Contextual information used in our system is defined at three hierarchical levels – atomic level, – composite level, – and high level as shown in Figure 3.5.

The atomic level collects all atomic context raw-data from different hardware sources, software sources, and the user interaction. The atomic context elements consists of basic information describing the learner (i.e. preferred language, previously conducted learning interactions), the used device (i.e. operating system, current available memory), the surrounding environment (i.e. network bandwidth, location) and the task at hand (i.e. current learner's query, learning goal). Some of these atomic context elements are sensed by software and hardware sensors and others are retrieved from user's input, user's profile, and used device. At this level, contextual information has its own static or dynamic attribute. Some of the contextual information is static, such as screen resolution of a specific handheld device, learner's birth date, gender, and preferred language(s). The other contextual information is dynamic such as network bandwidth, user's location, and current available memory. In addition, data type of raw-data is divided to quantized and non-quantized contexts whose values are numeric, Boolean, and literals, and most of which are time-stamped.

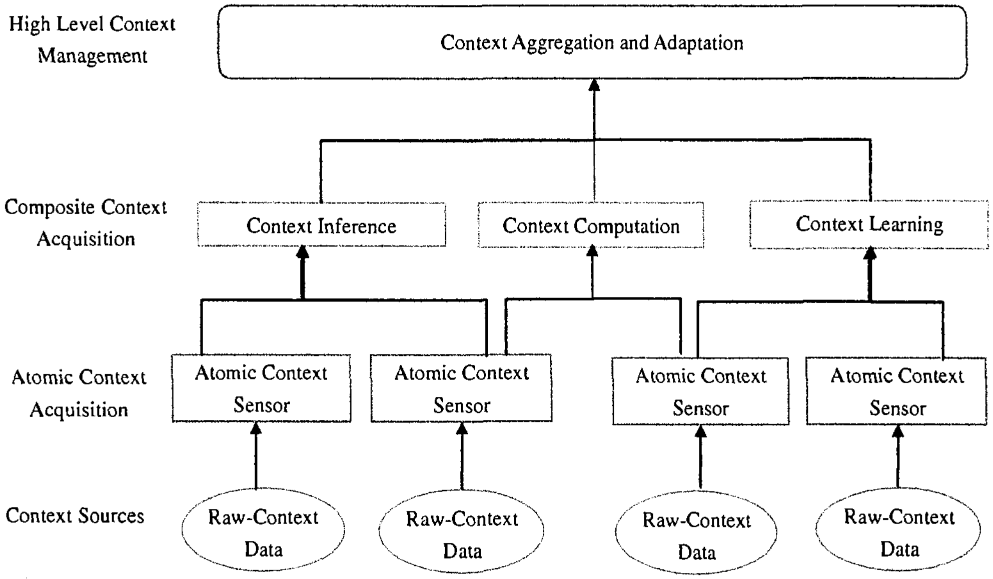


Figure 3.5 Context Acquisitions and Management

At the composite level, the system uses inference, computation, and learning techniques for translating all atomic contexts raw-data into symbolic values. The computed composite context is calculated from the atomic context elements. The inferred composite context is derived by inference using the Rule-Based Inference Engine written in SWRL. Ideally, all dynamic contextual-changes need to be fed to the system as they occur. However, the process of continuously sensing and updating the dynamic atomic context elements is time and resource consuming, especially in a mobile computing environment where system resources are very expensive. To solve this problem, the system adopts a learning technique where it senses precise values at some specific points in time and it predicts the approximate symbolic value between the sensing points. Example of computed composite context is user's age that can be computed from user's date of birth. Another example is interaction time that can be computed from interaction begin-time and end-time. Example of inferred composite context is inference of media type that can be played by a handheld device based on current network bandwidth. Example of learned context is translating current wireless network bandwidth to some symbolic meaningful value using fuzzy logic.

At the High level is context aggregation and adaptation where system divides context into four context groups: C_L is learner context; C_D is device context; C_E is environment context; and C_A is activity context. Learner context contains learner profile such as learner ID, authentication information, and preferred languages. Device context is the main source for determining the software and hardware capabilities of used devices. The software information consists of operating system, support languages, support media type for a particular mobile device. The hardware information consists of screen resolution, memory size, and display type and so on. Environment context is the main source for provisioning learner's surrounding context information that includes current bandwidth, current location, and used wireless network and so on. Activity context deals with accessed services, consumed learning resources, adopted learning sequence, and domain-knowledge management. The formal definition of these four context groups is defined as follows:

Context Aggregation = {Learner, Device, Environment, Activity}

T 1: Learner; Concepts ={Learner ID, Authentication Info, Preferred languages , Covered Concept, Consumed learning resource}

T1.1 Authentication Info; Concepts = {Username, Password}

T2: Device; Concepts = {Device ID, Software, Hardware}

T2.1: Software; Concepts = {Operating system, Support languages, Support media type, Run application}

T2.1.1 Operating system; Concepts = {Palm, Windows Mobile, RIM, GPE, OPIE, Symbian, Linux, Windows...}

T2.1.2 Support languages; Concepts = {English, France, German, ... }

T2.1.3: Support media type; Concepts = {Text, Image, Video}

T2.1.4 : Run application ; Concepts = {Word, PowerPoint, JPEG ...}

T2.2: Hardware; Concepts ={Display type; Keyboard type, Max bandwidth, Available memory, Screen resolution}

T2.2.1: Display type; Concepts = {Normal, Touch}

T2.2.1: Keyboard type; Concepts ={Virtual, Physical, Real}

T2.2.3: Screen Resolution; Concepts = {High. Width}

T3: Environment; Concepts = {Current bandwidth, Current Location, Sensed time, Wireless network}

T3.1: Location; Concepts = {Longitude, Latitude}

T3.2: Sensed Time; Concepts = {Current(yyyy : mm : dd ; hh :mm)}

T4: Activity; Concepts = {Activity ID, Query, Learning path}

T4.1 Query; Concepts = {Keyword}

T4.1: Learning path ; Concepts = {Is-a, Prerequisite, Necessary part of, Part of}

Table 3.1 shows some of the context data used in the system. These are identified according to the four context groups.

Source ID (Type)	**Feature ID (Type)**	**Value**	**Symbolic(Probability)**
1 (Device)	1 (Screen Resolution)	Identifier	{High, Width} & N/A
1 (Device)	2 (OS)	Identifier	{Linux, Symbian ...} &N/A
1 (Device)	3 (Media)	Identifier	{Text, Image, Video} & N/A
2 (Environment)	1.1 (Network: Bandwidth)	Kbps	{Low, Medium, High} & N/A
2 (Environment)	1.2 (Network: Latency)	Boolean	{True, False}
2 (Environment)	1.3 (Network: Security)	Boolean	{True, False}
2 (Environment)	2 (Location)	(X, Y)	{Stationary, On the Move}
2 (Environment)	3 (Time)	Identifier	{Current(yyyy:mm:dd; hh:mm)}
2 (Environment)	4 (Situation)	Boolean	{True, False}
3 (Learner)	1 (Authentication Info)	Boolean	{True, False}
3 (Learner)	2 (Languages)	Identifier	{English, French ...} & N/A
3 (Learner)	3 (Preference)	Identifier	{Name, ID, Address...}
4 (Activity)	1 (Query)	Identifier	{keyword, domain} & N/A

Table3.1 Context Definition

3.3 Ontology-based Context Modeling

The focus of current research work in the area of context-awareness is ontology-based context acquisition and management [41-44]. For instance, Yang [30] adopted two types of context ontologies: learner ontology and service ontology for collecting contextual information. These two ontologies are employed by the system to build a context-aware ubiquitous learning environment that can fully support the needs of peer-to-peer collaborative learning communities. CAMUS context model [28] used ontologies to formally describe contextual information related to agent, environment, device, location, and times. Their context model was used for the home domain and shows how it entails implicit reasoning. The system uses ontology to model and manage high level contexts at the semantic level to personalize the learning. In particular, it defines contextual information using a global ontology space that includes four interrelated sub-ontologies – learner ontology – activity ontology – device ontology – and environment ontology. In addition, domain ontology is used to define the subject domain area to be taught. The global ontology space describing knowledge about all context components is incremented with the domain ontology knowledge, and used as a unified knowledge base for system reasoning. As shown in Figure 3.6, the five ontologies are blended along the many properties that link various classes used by these ontologies. For example:

- The properties *HasCovered* and *ConsumedLearningResource* relate the *Learner* class to *Concept* and *LearningResource* class respectively. These relations are useful to track already covered concepts and consumed learning resources by a learner.

- The property *ConductedLearningActivity* relates the *Learner* class to *Learning Activity* class. It is used to help the system infer and retrieve all previously conducted learning interactions for a particular learner.

- Activity ontology and device ontology are linked through property *UsedDevice.*

- The environment ontology and learner ontology are linked by *HasSurroundingEnvironment* and *LocatedIn* properties which relate the *Learner* class to *Environment* and *Location* classes respectively.

The following sections describe in details each of these ontologies as well as the relationships between them.

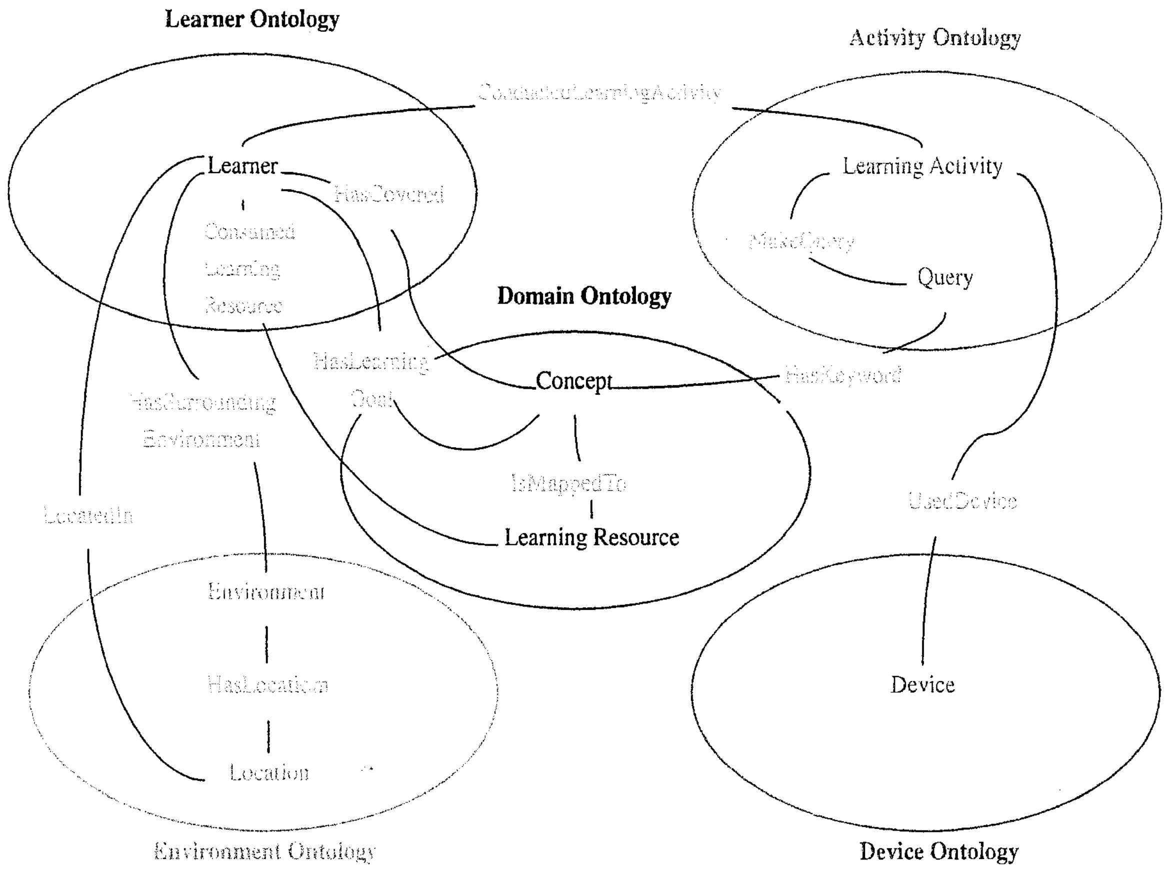


Figure 3.6 Global Ontology Space

3.3.1 Domain ontology

Domain ontology is used to represent and organize existing knowledge for a specific subject domain. It is expressed in terms of a hierarchy of subject topics, each of which is described by a set of concepts and their relationships. Figure 3.7 presents a hypothetical ontology for C++ Programming designed for our system. The figure includes a number of concepts and many instances of the following four ontology based relationships: *prerequisite, part-of, necessary part-of* and *is-a* [35].

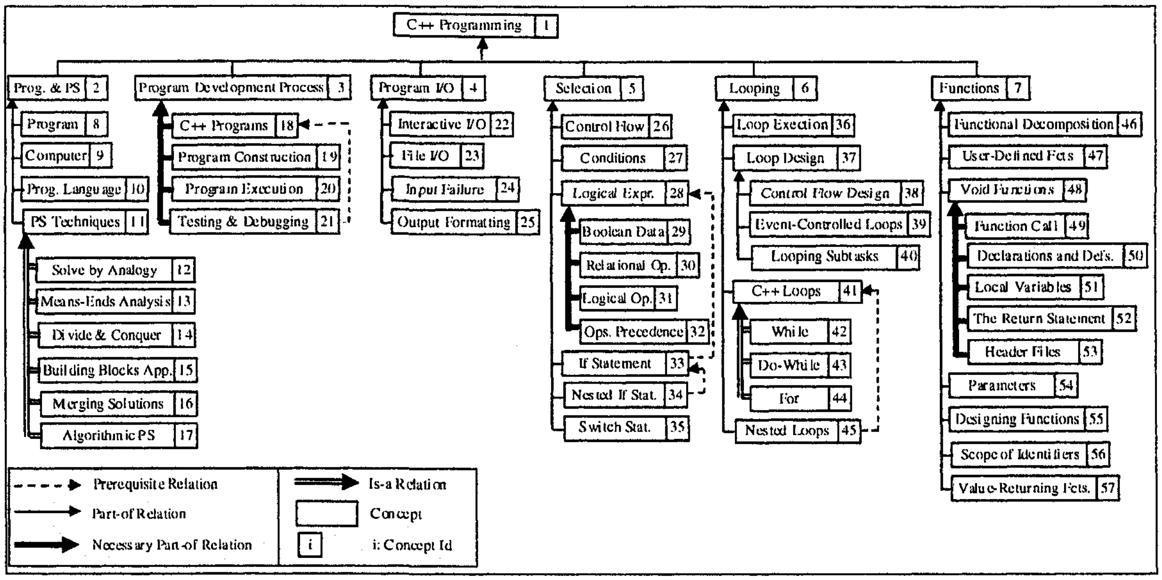


Figure3.7 Domain Ontology for C++ Programming

Figure 3.8 illustrate the formal structure and relationships used to define a domain ontology. The core class in the domain ontology is class *Concept* that can be used to represents all concepts shown in Figure 3.7. The relations *prerequisite, necessary part-of, part-of* and *is-a* describe prerequisite knowledge, core knowledge, related knowledge and similar knowledge between the various concepts respectively. The property *IsMappedTo* relates the *Concept* class to *Learning Resource* class. The properties *HasType, ExpressedIn* ,and *RunsOn* relate an individual of class *Learning Resource* to its attributes *Media Type, Language(s),* and *OS* respectively. These properties along with *HasKeyword* property, which associates keywords input by the learner to most related ontology concepts, are very useful for retrieving learning resources by mapping their metadata to ontology concepts, thus allowing resources sharing. The property *HasCovered* relates the covered domain ontology concepts to individuals of class Leaner. The *Concept* class and *Learner* class are linked through property *HasLearningGoal* that represents the learning goal for a particular learner. It should be noted that the literal *D* means domain, and literal *R* means range of the relationship in the following ontology figures.

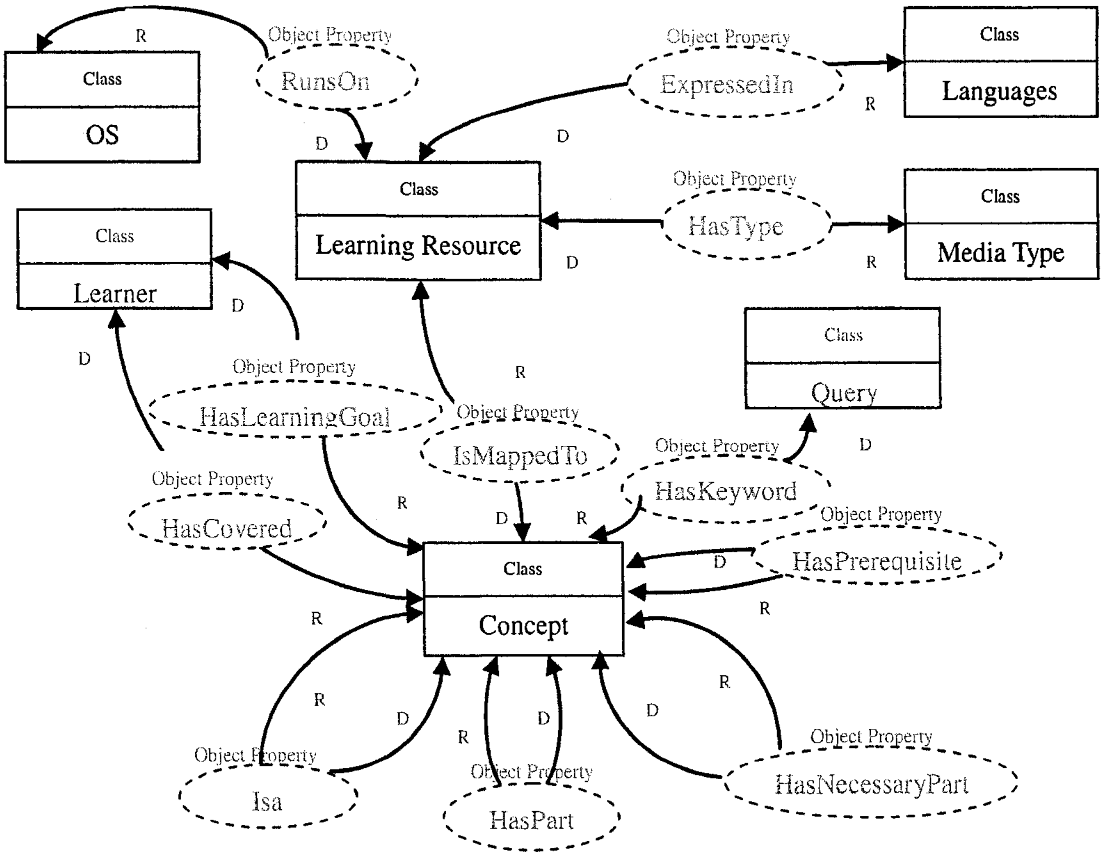


Figure3.8 Domain Ontology

3.3.2 Learner ontology

Learner ontology is an important part of our context model for representing contextual knowledge about the learner. This knowledge is organized into ontology concepts and relationships and used to map different contextual learner attributes onto service invocations, thus, enabling the system to discover, adapt, and deliver the most relevant learning resources in response to queries made by the learner. Figure 3.9 shows the structure and relationships used in this study to define a learner ontology.

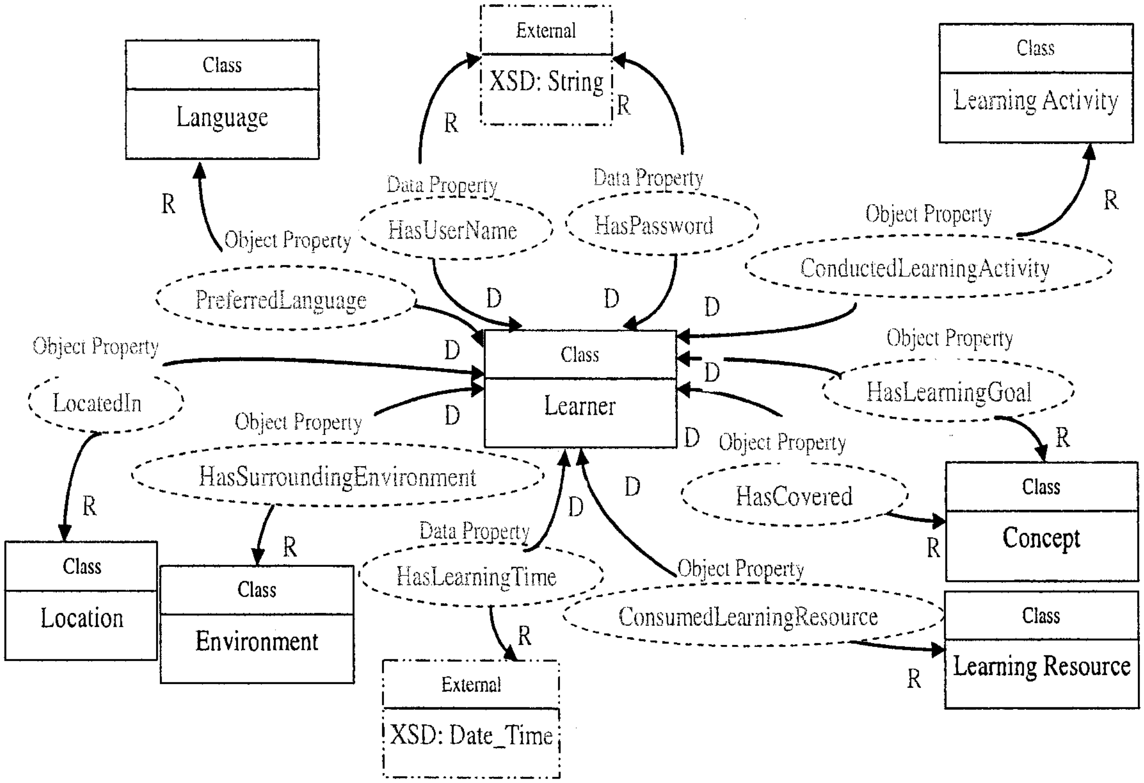


Figure 3.9 Learner Ontology

The property *PreferredLanguage* denotes learner preferred language(s) such as English, French, etc. The data properties *HasUserName* and *HasPassword* relate individuals of *Learner* class to their identification and authentication information. The properties *HasSurroundingEnvironment* and *LocatedIn* relate individuals of *Learner* class to *Environment* and *Location* classes, part of environment ontology respectively. These relationships are useful for retrieving learner's current environment (i.e. bandwidth), learner's location, and to infer new metadata for future system-centric adaptation. Other two important properties are *HasCovered* and *ConsumedLearningResource* which relate individuals of *Learner* class to the covered concepts and consumed learning resources in the domain ontology. These relations are useful to track already covered concepts and consumed learning resources by the learner and plan learning path for future learner-centric adaptation. The property *ConductedLearningActivity* relates the *Learner* class to *LearningActivity* class in the activity ontology. This relationship can help the system to infer and to retrieve all previously conducted learning interactions for a particular learner.

3.3.3 Device ontology

Device ontology is used to represent knowledge about the learner's used device(s). This knowledge is used for tracking the main characteristics of the used device in order to retrieve adaptive learning resourses for that particuar device. The device ontology includes knowledge related to both software-centric context and hardware-centric context. The software-centric context is classified into support languages, support media type, operation system, and software applications. The hardware-centric context is classified into device type, display type, keyboard type, max bandwidth, available memory, network adaptor, and screen resolution. Figure 3.10 shows the structure and relationships used by device ontology. The property *UsedDevice* relates *Device* class to *LearingActivity* class in activity ontology. This relationship is used to track the device used by a learner during a specific learning activity. It should be noted that symbol *1 means the range of the relationship is restricted to one. For instance, a device can only have one display type.

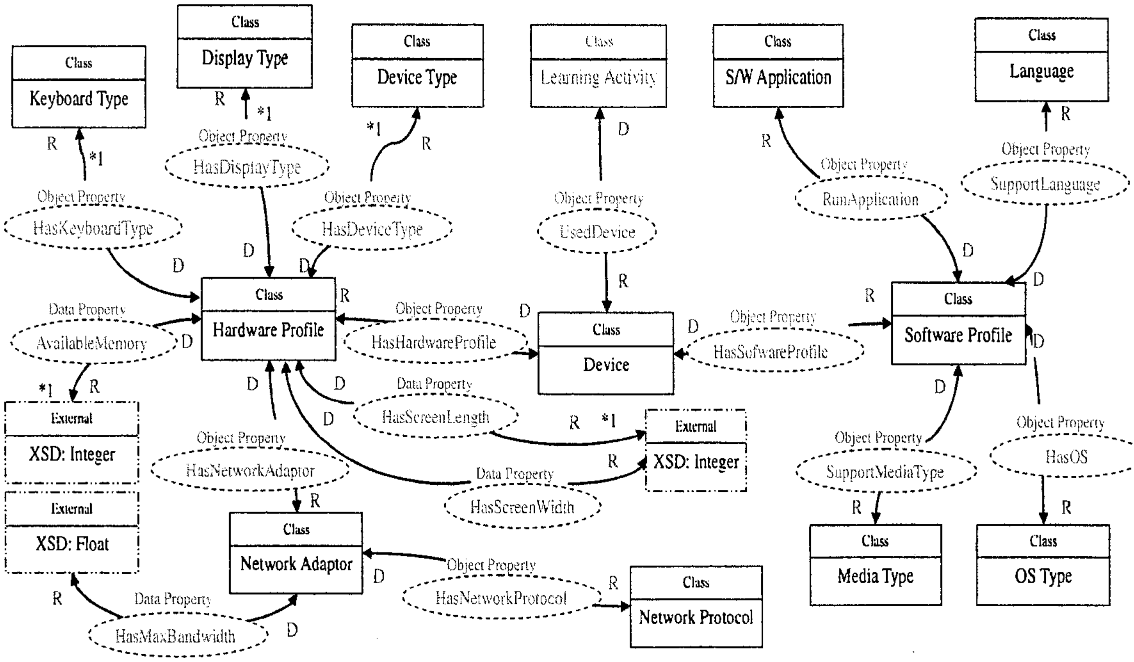


Figure3.10 Device Ontology

3.3.4 Environment ontology

Environment ontology is used to describe the knowledge about learner's surrounding environment. This knowledge consists of temporal and spatial contextual features, as well as network and security features. Figure 3.11 shows the structure and relationships for the environment ontology. The properties *HasWirelessNetworkType, IsSecured, HasBandwidth* denote that the learner is connected through that particualr wireless network type, with a specific security status, and a specific current bandwidth respectively. These contextual elements are very crucial for inferring and adjusting learning content that is compatible in terms of size, media-type, and privacy, with the technological set-up that characterizes the surrounding environment. The property *HasSurroundingEnvironment* associates learner's surrounding environment to individuals of *Learner* class. The property *LocatedIn* relates learner's location to individuals of *Learner* class.

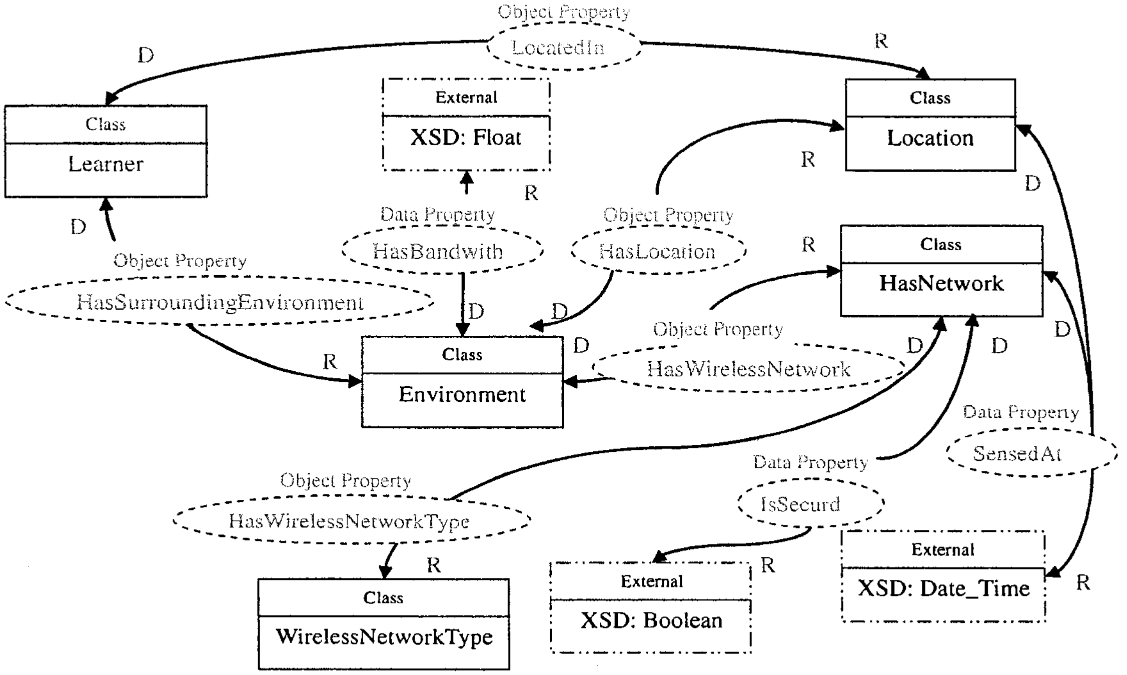


Figure3.11 Environment Ontology

3.3.5 Activity ontology

Activity ontology is used to describe knowledge about a learning activity. This knowledge records the learner's interaction with a specific handheld device in a period of time. Figure 3.12 shows the main concepts and their relationships for activity ontology. The *Learning Activity* class is the core class in the activity ontology. The properties *Begin-time* and *End-time* describe the time period of the learner's interaction. These relationships can be used to retrieve, for instance, the learner's previous interactions in case of network interruption. The property *HasActivityID* provides identification information to an individual of *LearningActivity* class. It should be noted that a learner can only have one activity ID at any one time. This activity ID should be terminated when the learner logs out. The property *MakeQuery* allows inferring all queries made during a learning interaction with the system. The property *HasKeyword* relates *Query* class to *Concept* class of domain ontology. Query keywords are directly mapped to domain ontology concepts. Learners should use ontology vocabulary while composing their queries. The *Learning Activity* class is related to *Device* class along the property *Used Device.*

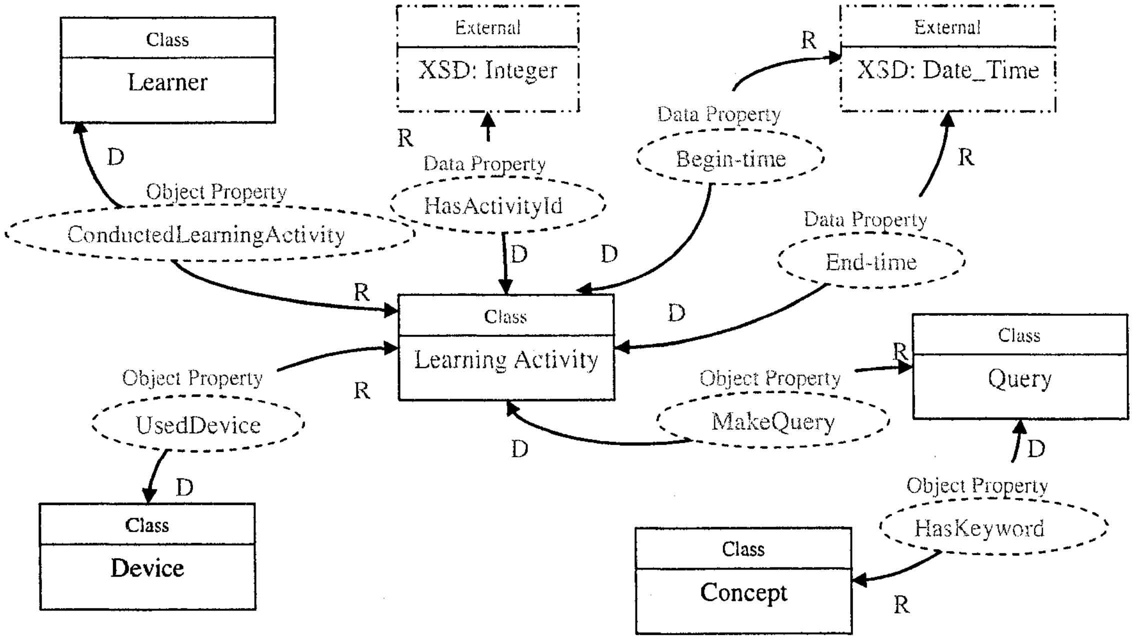


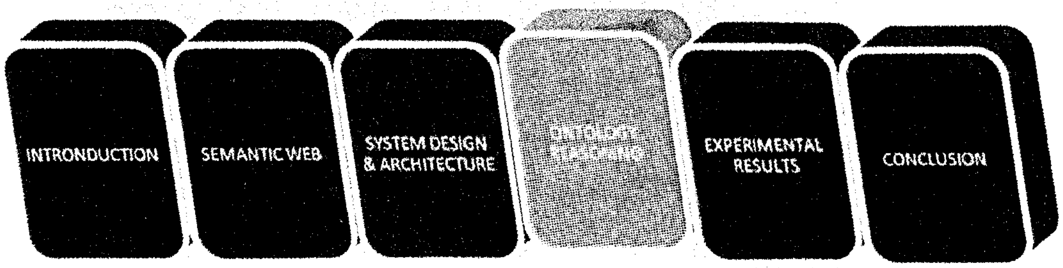
Figure 3.12 Activity Ontology

Here is a fragment of OWL description for the global ontology space as generated by Protégé. This is used to define the classes, properties, and a specific learner.

```
<owl:Class rdf:about="#Learner">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="HasSurroundingEnvironment"/>
    </owl:onProperty>
    <owl:allValuesFrom><owl:Class rdf:about="#Environment"/></owl:allValuesFrom>
   </owl:Restriction>
 </rdfs:subClassOf>
  <owl:disjointWith> <owl:Class rdf:about="#Environment"/></owl:disjointWith>
  <HasConsumed><owl:Class rdf:ID="LearningResource"/></HasConsumed>
  <rdfs:subClassOf><owl:Restriction>
    <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Username
     </owl:hasValue>
    <owl:onProperty>
     <owl:DatatypeProperty rdf:about="#AuthenticateBy"/>
    </owl:onProperty>
   </owl:Restriction></rdfs:subClassOf>
  ...
  </owl:Class>

  <owl:Class rdf:ID="LearnerID"><rdfs:subClassOf rdf:resource="#Learner"/></owl:Class>

   <owl:ObjectProperty rdf:about="#IsConsumedBy">
    <rdfs:range rdf:resource="#Learner"/>
    <owl:inverseOf><owl:ObjectProperty rdf:about="#HasConsumed"/></owl:inverseOf>
    <rdfs:domain rdf:resource="#LearningResource"/>
   </owl:ObjectProperty>

  <LearnerID rdf:ID="Irene">
     <HasConsumed>
       <LearningResource rdf:ID="Computer1">
        <Mappes rdf:resource="#Computer"/>
        <IsConsumedBy rdf:resource="#Irene"/>
       </LearningResource>
     </HasConsumed>
       <HasCovered rdf:resource="#Computer"/>
 </LearnerID>
```

CHAPTER 4

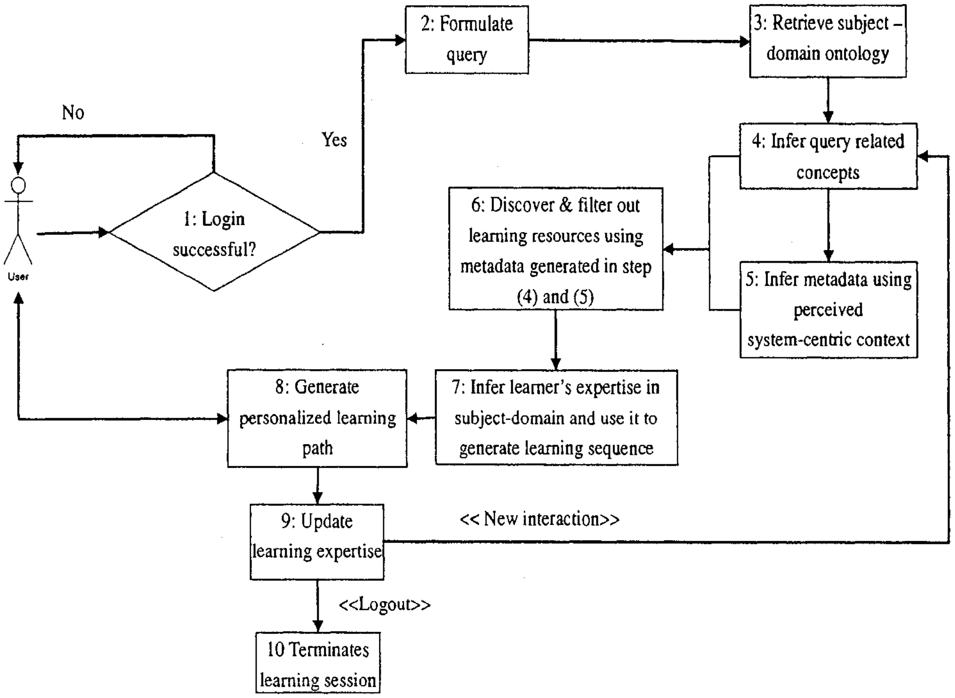
ONTOLOGY REASONING



This chapter describes the ontology reasoning and learning strategies used to personalize mobile learning services. Ontology-based reasoning consists of system-centric adaptations and learner-centric adaptations. The system-centric adaptations ensure searched learning resources are exercisable on the used handheld device, while the goal of learner-centric adaptations is to build a learning path that suits the learner's background and current activity. This chapter first presents the processing steps in a typical mobile learning scenario. It then describes the various adaptations employed in this study.

4.1 Processing Steps in a Typical Learning Scenario

This section overviews the main processing steps in a typical learning scenario. The high level context is fed to the ontology reasoning engine in order to personalize learning services based on the learner context, device context, environment context, and activity context. This adaptation process is achieved in two successive stages – system-centric adaptation – and learner-centric adaptation. System-centric adaptation is based on device ontology and environment ontology. It consists in applying a set of rules to infer the system-centric metadata (i.e. media type, search language) for use in service discovery process. Learner-centric adaptation is however based on learner ontology, activity ontology, and domain ontology. It consists of applying a set of ontological rules to infer metadata that can be used to customize the learning path. The sequence of steps given below illustrates the personalization process in a typical learning scenario where a learner wants to acquire knowledge in a specific learning domain. This learning scenario is also depicted graphically in Figure 4.1.



2: Formulate query
3: Retrieve subject - domain ontology
No
Yes
4: Infer query related concepts
6: Discover & filter out learning resources using metadata generated in step (4) and (5)
1: Login successful?
User
5: Infer metadata using perceived system-centric context
8: Generate personalized learning path
7: Infer learner's expertise in subject-domain and use it to generate learning sequence
9: Update learning expertise
<< New interaction>>
<<Logout>>
10 Terminates learning session

Figure 4.1 Processing Steps in a Typical Learning Scenario

1. When the learner logs in, his background, preferences, and previous learning activity are retrieved.
2. The learner uses the domain ontology concepts to query the system.
3. The subject-domain ontology related to the learner's query is retrieved.
4. Based on the learner's query, the system infers the related ontology concepts and identifies those concepts that are part of similar knowledge, prerequisite knowledge, core-knowledge, and related knowledge using *Is-a, HasPrerequisite, HasNecessaryPartOf,* and *HasPartOf* properties respectively.
5. Next, the system uses the perceived device and environment atomic context elements to infer metadata that adapts the search for those learning resources that are suitable for the system-centric context.

6. The metadata generated in (4) that is associated with the related domain ontology concepts, and the system-centric metadata generated in (5), are then used to discover and filter out learning resources stored in various learning repository based on system-centric context.
7. The system will then determine the learner's expertise in the subject-domain (i.e. tacit knowledge) by inferring previous learning activities, covered concepts, adopted learning paths, and consumed learning resources. This knowledge is used to build a personalized learning sequence by removing already covered learning concepts, learning resources, and learning paths. Thus, the newly constructed learning sequence consists of optimized system-centric learning resources related to knowledge that has not been covered by the learner so far.
8. The personalized learning sequence is then provided to the learner for navigation.
9. Based on the newly selected concept, learner's expertise is automatically updated and the personalized learning path is re-adjusted by resuming processing from step (4).
10. The learning activity terminates when either the learner logs out, or when all domain concepts are covered.

4.2 System-Centric Adaptation

The system-centric adaptations aim at filtering out those learning resources that are runnable on the used handheld device. This is achieved through inference of system-centric metadata based on perceived device and environment atomic context elements. The system-centric adaptation achieves its functionality in three steps: (1) media type and file size adaptation, (2) search language adaptation and, (3) other resource-centric adaptation. The diagram shown in Figure 4.2 describes the logical steps to select the media type of retrieved learning resources to make sure they are browsable on the used device. When the learner logs in, the system will first sense the used network adaptor and retrieves its connection speed. Connection speed is an attribute that is straightforward to obtain and it is typically the maximum theoretical speed for the used wireless adapter [45]. Knowing the type of the network connection, such as IEEE 802.11 wireless LAN or General Packet Radio Service (GPRS - wireless WAN), gives our reasoning engine the insight that allows it to make some adaptation choices related to media-type and size of resources to be retrieved. This is achieved by taking into account the available bandwidth and device features. For example, if the network connection is IEEE 802.11, the system will not keep sensing the network connection and will not make any restrictions on the type and size of media because the available bandwidth is large enough to handle all type of resources. However, if the sensed connection is GPRS, the system adapts the media-type and resource size based on the available bandwidth as explained below. For example for a GPRS connection, the maximum connection speed could be 48.0 kbps, however, the actual network bandwidth is usually less than that due to traffic on the network [45]. Ideally, the system should continuously sense the current network bandwidth and update the associated atomic context element whenever bandwidth change occurs. However, the process of continuously sensing and updating such dynamic bandwidth is time and resource consuming as it involves sending data packets through the network. To solve this problem, the system only sense the actual bandwidth at some points in time, and it uses a fuzzy logic approach in conjunction with SWRL rules to predict the available bandwidth between these points. Also, to reason with bandwidth, fuzzy logic translate the predicted current bandwidth into meaningful symbolic values such as low, medium, and high bandwidth as described below. Fuzzy logic is also used to predict the maximum data file size that can be communicated to the used device to avoid experiencing long delays. For instance, the system only search for learning resources with text type if a mobile device, operating on a GPRS network for instance, has a low bandwidth. However, it can accept learning resources of image or video type if the network bandwidth is high. The system-centric adaptations also check whether the operating system that is required to run the learning resource is similar to device's operating system for compatibility purpose.

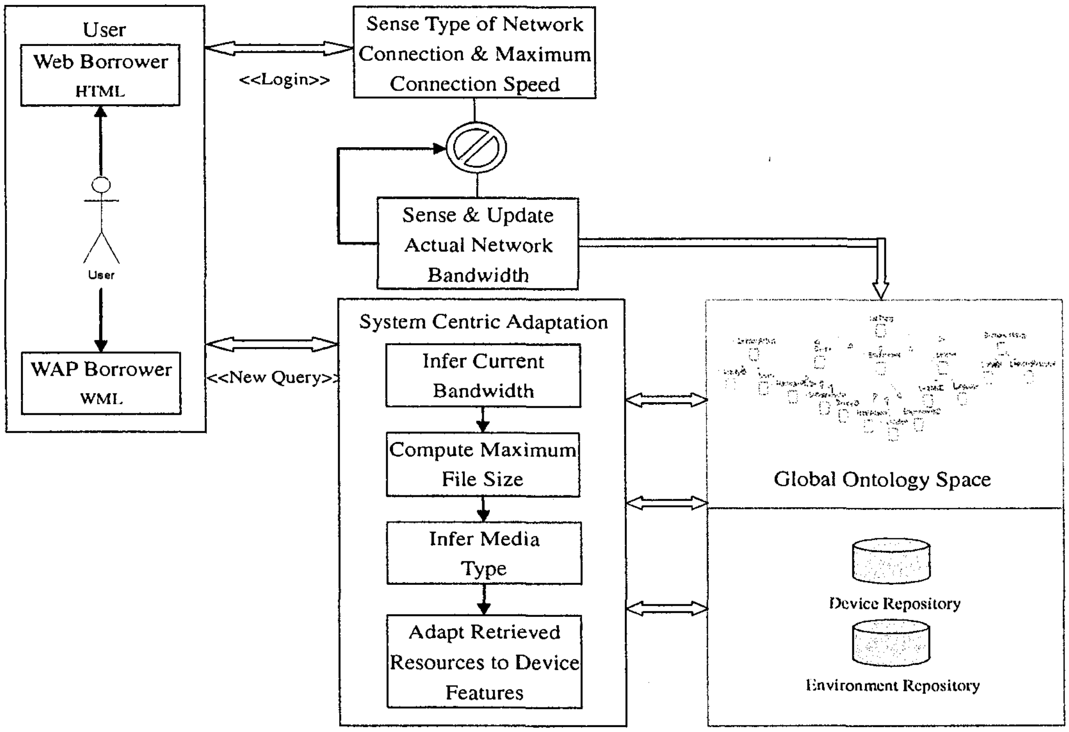

User
Web Borrower
HTML
User
WAP Borrower
WML
<<Login>>
<<New Query>>
Sense Type of Network Connection & Maximum Connection Speed
Sense & Update Actual Network Bandwidth
System Centric Adaptation
Infer Current Bandwidth
Compute Maximum File Size
Infer Media Type
Adapt Retrieved Resources to Device Features
Global Ontology Space
Device Repository
Environment Repository

Figure 4.2 System-Centric Adaptations

Fuzzy logic's symbolic values are used to describe the current network bandwidth and available device memory as described above. The system make uses of the fuzzy logic truth values in conjunction with SWRL rules to allocate symbolic value to the current network bandwidth. Figure 4.3 shows the way the system predicts the current network bandwidth using the fuzzy qualifying linguistic variables such as *Low*, *Medium*, and *High.* The symbol $\mu_A(x)$ represents a truth value that is between 0 and 1. This can be computed by Equation 4.1.

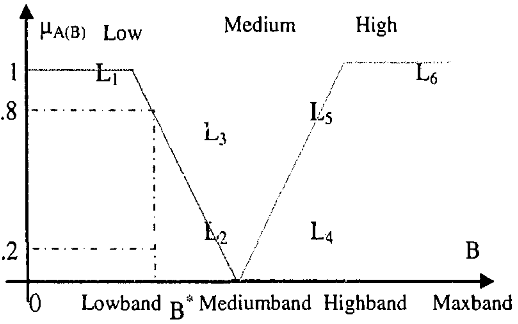


Figure 4.3 Membership Function for Bandwidth

$$
\begin{aligned}
&L_1 : \mu_A(b) = 1 \\
&L_2 : \mu_A(b) = -\frac{b - Mediumband}{Mediumband - Lowband} \\
&L_3 : \mu_A(b) = \frac{b - Lowband}{Mediumband - Lowband} \\
&L_4 : \mu_A(b) = -\frac{b - Highband}{Highband - Mediumband} \\
&L_5 : \mu_A(b) = \frac{b - Mediumband}{Highband - Mediumband} \\
&L_6 : \mu_A(b) = 1
\end{aligned}
\qquad \text{(Equation 4.1)}
$$

Table 4.1describes the SWRL rules that are used to infer the truth values of classified symbolic network bandwidth given in Equation 4.1. The property *UsedDevice*(?*a*,?*y*) relates an individual learner identified by his/her activity identifier *a* to his/her mobile device *y*. *TruthValueRule-1* and *TruthValueRule-2* are respectively related to L_1 and L_2 (Equation 4.1), and are used to infer the truth values associated to low network bandwidth. *TruthValueRule-3* and *TruthValueRule-4* are respectively related to L_3 and L_4 (Equation

4.1), and are used to infer the truth values for medium network bandwidth. Finally, *TruthValueRule-5* and *TruthValueRule-6* are respectively related to L_5 and L_6 (Equation 4.1), and are used to infer the truth values for high network bandwidth. The data properties *HasBandwidth* and *MaxBandwidth* represent respectively the current network bandwidth and maximum connection speed of used mobile devices.

SWRL **Rules**	**TruthValueRule-1** ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y,?Maxband) ∧ swrlb:multiply(?Lowband,?Maxband, 0.25) ∧ swrlb:lessThanOrEqual(?b, ?Lowband) → ProbLow(?y, 1.0) ∧NetworkBandwidth(?y, "Low") **TruthValueRule-2** ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y,?Maxband) ∧ swrlb:multiply(?Lowband,?Maxband,0.25) ∧ swrlb:multiply(?Mediumband, ?Maxband, 0.5) ∧ swrlb:greaterThan(?b, ?Lowband) ∧ swrlb:lessThanOrEqual(?b, ?Mediumband) ∧ swrlb:subtract(?z1, ?Mediumband, ?b) ∧ swrlb:subtract(?z2, ?Mediumband, ?Lowband) ∧ swrlb:divide(?z, ?z1, ?z2) → ProbLow(?y, ?z) ∧ NetworkBandwidth(?y, "Low") **TruthValueRule-3** ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y,?Maxband) ∧ swrlb:multiply(?Lowband,?Maxband,0.25) ∧ swrlb:multiply(?Mediumband, ?Maxband, 0.5) ∧ swrlb:greaterThan(?b, ?Lowband) ∧ swrlb:lessThanOrEqual(?b, ?Mediumband) ∧ swrlb:subtract(?z1, ?b, ?Lowband) ∧ swrlb:subtract(?z2, ?Mediumband, ?Lowband) ∧ swrlb:divide(?z, ?z1, ?z2) → ProbMedium(?y, ?z) ∧ NetworkBandwidth(?y, "Medium") **TruthValueRule-4**

	ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y, ?Maxband) ∧ swrlb:multiply(?Highband,?Maxband,0.75) ∧ swrlb:multiply(?Mediumband, ?Maxband, 0.5) ∧ swrlb:greaterThan(?b, ?Mediumband) ∧ swrlb:lessThanOrEqual(?b, ?Highband) ∧ swrlb:subtract(?z1, ?Highband, ?b) ∧ swrlb:subtract(?z2, ?Highband, ?Mediumband) ∧ swrlb:divide(?z, ?z1, ?z2) → ProbMedium(?y, ?z) ∧ NetworkBandwidth(?y, "Medium")
	TruthValueRule-5 ActivityID(?a) ∧UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y,?Maxband) ∧ swrlb:multiply(?Highband,?Maxband,0.75) ∧ swrlb:multiply(?Mediumband, ?Maxband, 0.5) ∧ swrlb:greaterThan(?b, ?Mediumband) ∧ swrlb:lessThanOrEqual(?b, ?Highband) ∧ swrlb:subtract(?z1, ?b, ?Mediumband) ∧ swrlb:subtract(?z2, ?Highband, ?Mediumband) ∧ swrlb:divide(?z,?z1,?z2) → ProbHigh(?y, ?z) ∧ NetworkBandwidth(?y, "High")
	TruthValueRule-6 ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?y, GPRS) ∧ MaxBandwidth(?y,?Maxband) ∧ swrlb:multiply(?Highband,?Maxband,0.75) ∧ swrlb:greaterThanOrEqual(?b, ?Highband) → ProbHigh(?y, 1.0) ∧NetworkBandwidth(?y, "High")

Table 4.1 SWRL Rules for Truth Value

The following example illustrates the way the system applies the SWRL rules shown in Table 4.1 in a real-life scenario. For instance, let's assume that *Irene* is using mobile device MotoW270 with a maximum connection speed of 32.0 kbps. Let's also assume that the value (x^*), the previously sensed actual network bandwidth, is found to be around 18.0 kbps, that is fluctuating between medium to high bandwidth with relation to the maximum connection speed (see Figure 4.3). When *TruthValueRule-2* and

TruthValueRule-3 rules are applied, facts *A2* and *B2* are inferred, resulting into the addition of four statements to the list of facts as shown in Table 4.2. These new facts reveal the probabilities for the predicted current bandwidth which were found to be 0.75 for medium bandwidth and 0.25 for high bandwidth as shown in Table 4.2.

	Ontology related Facts	**Inferred Facts**

Facts	*A1)* **TruthValueRule-4**
	ActivityID(Irene)
	UsedDevice(Irene, MotoW270)
	HasBandwidth(MotoW270, 18.0)
	HasNetworkAdaptor(MotoW270, GPRS)
	MaxBand(MotoW270, 32.0)
	swrlb:multiply(?Highband,32.0,0.75)
	swrlb:multiply(?Mediumband, 32.0, 0.5)
	swrlb:greaterThan(18.0, 16.0)
	swrlb:lessThanOrEqual(18.0, 24.0)
	swrlb:subtract(?z1, 24.0, 18.0)
	swrlb:subtract(?z2, 24.0, 16.0)
	swrlb:divide(?z, 6.0, 8.0)
		A2)
		ProbMedium(MotoW270, 0.75)
		NetworkBandwidth (MotoW270, "Medium")

	B1) **TruthValueRule-5**
	ActivityID(Irene)
	UsedDevice(Irene, MotoW270)
	HasBandwidth(MotoW270, 18.0)
	HasNetworkAdaptor(MotoW270, GPRS)
	MaxBand(MotoW270, 32.0)
	swrlb:multiply(?Highband,32.0,0.75)
	swrlb:multiply(?Mediumband, 32.0, 0.5)
	swrlb:greaterThan(18.0, 16.0)
	swrlb:lessThanOrEqual(18.0, 24.0)
	swrlb:subtract(?z1, 18.0, 16.0)
	swrlb:subtract(?z2, 24.0, 16.0)
	swrlb:divide(?z,2.0,8.0)
		B2)
		ProbHigh(MotoW270, 0.25)
		NetworkBandwidth (MotoW270, "High")

Table 4.2 Instance for Truth Value

The inferred probabilities of current bandwidth are then used to infer the maximum allowable data file size. It should be noted that the resource size is checked for efficiency purpose, as it is not practical to consider a large resource (i.e. Mbytes) if the used device operates on a low bandwidth (i.e. few kbps). So, based on the response time obtained in experiments done on real mobile devices, the system identified some threshold values for resource sizes that can typically be used for specific bandwidth ranges. Based on these thresholds, it has adopted the following assumptions. If a mobile device has a connection speed less than 32.0 kbps, it should not consider resources that exceed 500.0Kbytes. In other words, system does not tolerate response times longer than 15 seconds. Similarly, if the connection speed was between 32.0 kbps to 66.0 kbps, then resources over 1Mbytes should not be considered. It should be noted that the maximum tolerable response time can be easily modified to accommodate learners with more or less restrictive time constraints. The system uses the predicted symbolic values associated with the current available bandwidth to predict the maximum allowable size of resources. Figure 4.4 shows the membership function for file size. The three fuzzy sets *Low, Medium,* and *High* describing predicted network bandwidth are used as an input space in the fuzzy system to predict the maximum allowable file size. It also defines three fuzzy sets *Small, Medium,* and *Large* as the output space (file size) as shown in Figure 4.4. Note that system uses *SmallSize, MediumSize,* and *LargeSize* to refer to center average values for small fuzzy set, medium fuzzy set, and large fuzzy set respectively as shown in Figure 4.4.

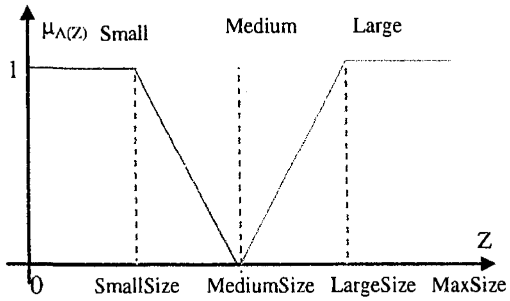


Figure 4.4 Membership Function for File Size

The rule base for our fuzzy logic system is given below.

R^1 : IF network bandwidth (B) is *Low* THEN file size (Z) is *Small.*

R^2 : IF network bandwidth (B) is *Medium,* THEN file size (Z) is *Medium.*

R^3 : IF network bandwidth (B) is *High,* THEN file size (Z) is *Large.*

Then, the crisp output (file size) from the fuzzy system with singleton fuzzifier, product inference engine, center average defuzzifier, and the rule base R^k, is given by Equation 4.2.

$$Z^* = \frac{\sum_{i=1}^{3} \overline{Z}_i * \mu_{A^i}(B^*)}{\sum_{i=1}^{3} \mu_{A^i}(B^*)} = \frac{SmallSize * \mu_{Low}(B^*) + MediumSize * \mu_{Medium}(B^*) + LargeSize * \mu_{Large}(B^*)}{\mu_{Low}(B^*) + \mu_{Medium}(B^*) + \mu_{Large}(B^*)}$$

Equation 4.2

Table 4.3 defines the SWRL rule for Equation 4.2. In *FileSizeRule-1*, the data properties *ProbLow, ProbMedium,* and *ProbHigh* are those obtained from Table 4.1. To show how these rules are applied in our system, this section provides a real-life scenario. Let's assume that learner is using a GPRS connection with a maximum connection speed of 32.0 kbps. This connection speed delimits the maximum file size to 500.0Kbytes as described above. These assumptions are represented by fact *A1* in Table 4.3. When rule *FileSizeRule-1* is applied, fact *A2* is inferred, resulting into the addition of statement *FileSize(MotoW270, 281.25)* to the list of facts. Therefore, since the previously sensed network bandwidth was 18.0 kbps, our system chooses not to exchange data files over 281Kbytes as deduced from the set of inferences shown in Table 4.3.

SWRL Rules	**FileSizeRule-1** ActivityID(?a)∧UsedDevice(?a, ?y)∧ProbLow(?y, ?Tl)∧ ProbMedium(?y, ?Tm)∧ProbHigh(?y, ?Th)∧MaxSize(?y, ?Maxsize) ∧swrlb:multiply(?Lowsize, 0.25, ?Maxsize)∧ swrlb:multiply(?Mediumsize, 0.5, ?Maxsize)∧ swrlb:multiply(?Largesize, 0.75, ?Maxsize)∧ swrlb:multiply(?l, ?Lowsize, ?Tl)∧ swrlb:multiply(?m, ?Mediumsize, ?Tm)∧ swrlb:multiply(?h, ?Largesize,?Th)∧swrlb:add(?z1,?l,?m, ?h)∧ swrlb:add(?z2, ?Tl, ?Tm, ?Th)∧swrlb:equal(?z2, 1)∧ swrlb:divide(?z, ?z1, ?z2) → FileSize(?y, ?z)	
Facts	**Ontology related Facts**	**Inferred Facts**

	A1)
	ActivityID(Irene)UsedDevice(Irene, MotoW270)
	ProbLow(MotoW270, 0.0)
	ProbMedium(MotoW270, 0.75)
	ProbHigh(MotoW270, 0.25)
	MaxSize(MotoW270, 500)
	swrlb:multiply(?Lowsize, 0.25, 500.0)
	swrlb:multiply(?Mediumsize, 0.5, 500.0)
	swrlb:multiply(?Largesize, 0.75, 500.0)
	swrlb:multiply(?l, 125.0, 0.0)
	swrlb:multiply(?m, 250.0, 0.75)
	swrlb:multiply(?h, 375.0, 0.25)
	swrlb:add(?z1, 0.0, 187.5, 93.75)
	swrlb:add(?z2, 0.0, 0.75, 0.25)
	swrlb:divide(?z, 281.25, 1)
		A2)
		FileSize
		(MotoW270,281.25)

Table 4.3 SWRL Rule for File Size

Table 4.4 contains the SWRL rules used to select the media type of retrieved learning resources based on current bandwidth. The data properties *NetworkBandwidth* and *AvailableMemory* respectively represent the current bandwidth and available device memory. In *MediaRule-1*, the system sets the media type to text format when the current bandwidth is low. In *MediaRule2* the system sets the media type to text and image formats when the current bandwidth is medium, while in *MediaRule-3*, the system sets the media type to text, image, and video formats when the current bandwidth is high. The system will also adjust the maximum allowable file size, computed in Table 4.3, based on the device available memory. If the device memory size is smaller than the maximum allowable file size computed in Table 4.3, then *AllowedFileSizeRule-1* sets the maximum file size to the device memory size; otherwise the maximum file size remains unchanged as stated in *AllowedFileSizeRule-2.*

SWRL Rules	**MediaRule-1** ActivityID(?a)∧UsedDevice(?a,?y)∧NetworkBandwidth(?y,"Low") → HasMediaType(?y, Text) **MediaRule-2** ActivityID(?a)∧UsedDevice(?a, ?y)∧NetworkBandwidth(?y, "Medium")→HasMediaType(?y, Text)∧HasMediaType(?y, Image) **MediaRule-3** ActivityID(?a)∧UsedDevice(?a, ?y)∧NetworkBandwidth(?y, "High") →HasMediaType(?y,Text)∧ HasMediaType(?y,Image)∧ HasMediaType(?y,Video) **AllowedFileSizeRule-1** ActivityID(?a)∧UsedDevice(?a, ?y)∧FileSize(?y, ?Size)∧ AvailableMemory(?y,?MemorySize)∧ swrlb:lessThan(?MemorySize,?Size)→AllowedSize(?y,?MemorySize) **AllowedFileSizeRule-2** ActivityID(?a)∧UsedDevice(?a,?y)∧FileSize(?y,?Size)∧ AvailableMemory(?y, ?MemorySize)∧ swrlb:greaterThanOrEqual(?MemorySize, ?Size)→ AllowedSize(?y,?Size)

Table 4.4 SWRL Rules for Media Type & Allowable File Size

To show how the above rules are applied this section uses the previous scenario of learner *Irene* who is using device *MotoW270* operating at a bandwidth of 18.0Kbps to access the system services. Following the reasoning shown in Table 4.2, the system infers a bandwidth fluctuating between medium to high as shown in facts *A1* and *B1.* When applying *MediatypeRule-2* and *MediatypeRule-3,* facts *A2* and *B2* are respectively inferred and added to the list of facts. In *C1, AllowedFileSizeRule-2* is applied to compare the maximum allowable file size, inferred in Table 4.3, with the device available memory, leading to the addition of statement *AllowedSize (MotoW270, 281.25)* to the list of facts as shown in fact *C2*. As shown in Table 4.5, the system concludes that for this scenario, all types of media can be selected for delivery. It also concludes that these resources should not exceed a size of 281.25Kbytes for them to be ported on the used device, and to avoid long communication delays.

	Ontology related Facts	Inferred Facts

Facts	***A1)***
	ActivityID(Irene)
	UsedDevice(Irene,MotoW270)
	NetworkBandwidth
	(MotoW270,"Medium")
		A2)
		HasMediaType(MotoW270,Text)
		HasMediaType(MotoW270,Image)

	B1)
	ActivityID(Irene)
	UsedDevice(Irene, MotoW270)
	NetworkBandwidth
	(MotoW270,"Large")
		B2)
		HasMediaType(MotoW270,Text)
		HasMediaType(MotoW270,Image)
		HasMediaType(MotoW270,Video)

	C1)
	ActivityID(Irene)
	UsedDevice(Irene, MotoW270)
	FileSize(MotoW270, 281.25)
	AvailableMemory(MotoW270, 1024.0)
	swrlb:greaterThanOrEqual
	(1024.0, 281.25)
		C2)
		AllowedSize
		(MotoW270, 281.25)

Table 4.5 Instance for Media Type & File Size

Another system-centric adaptation considered in this study is to determine the language to be used by the search agent. *LanguageRule-1* in table 4.6 establishes a constraint represented by the relationship *SearchLannguage* between an activity ID and a language. The property $\Pr eferredLanguage(?a,?z)$ relates an ActivityID *a* to a preferred language *z*. The property $HasSupportLanguage(?y,?z)$ relates learner's handheld device *y* to it support language *z*. For instance, let's assume *French* is the preferred language for learner *Irene*. Let's also assume that *English* and *French* are languages supported by the used device *MotoW270.* When applying *LanguageRule-1,* as shown by *A1* in Table 4.6, system can infers *A2* that is *SearchLanguage (Irene, French),* confirming that *French* can be used by the agent as a search language because it is supported by the used device.

SWRL Rules	**LanguageRule-1** ActivityID(?a) ∧UsedDevice(?a,?y) ∧ PreferredLanguage(?a, ?z) ∧ HasSupportLanguage(?y,?z) → SearchLanguages(?a, ?z)	
Facts	**Ontology related Facts**	**Inferred Facts**

	A1)
	ActivityID(Irene)
	UsedDevice(Irene, MotoW270)
	PreferredLanguage(Irene,French)
	HasSupportLanguage(MotoW270, English)
	HasSupportLanguage(MotoW270, French)
		A2)
		SearchLanguages
		(Irene, French)

Table 4.6 SWRL Rule for Search Language

Finally, the last system-centric adaptation concerns the operating system required to run the learning resources. Table 4.7 shows the rules used for this type of adaptation as well as a scenario to illustrate such adaptation. In *SystemCentricRule-1*, it is shown that only those resources that can run on the device's operating system are considered. For instance, let us assume that the learning resource *C++ Loops* is expressed in *English* and it is of media type text. Its attributes are compatible with the learner's used mobile device *MotoW270.* These are represented by the facts *A1* in Table 4.7. When rule *SystemCentricRule-1* is applied, fact *A2* is inferred, resulting into the addition of statement *SystemCentric (Irene, C++Loops)* to the list of facts.

SWRL Rules	**SystemCentricRule-1** SearchLanguages(?y,?z) ∧ExpressedIn(?LR,?z) ∧ UsedDevice(?y, ?D) ∧ HasMediaType(?D,?b) ∧HapType(?LR,?b) ∧ HasOS(?D,?c) ∧RunsOn(?LR,?c)→ SystemCentric(?y, ?LR)	
Facts	**Ontology related Facts**	**Inferred Facts**

	A1)
	SearchLanguages(Irene,English)
	ExpressedIn(C++ Loops, English)
	UsedDevice(Irene, MotoW270)
	HasMediaType(MotoW270, Text)
	HapType(C++ Loops, Text)
	HasOS(MotoW270,Symbian)
	RunsOn(C++ Loops, Symbian)
		A2)
		SystemCentric
		(Irene,
		C++ Loops)

Table 4.7 SWRL Rule for System-Centric Adaptation

4.3 Learner-Centric Adaptation

Usually a single learning resource will not be enough for the learner to meet his learning goal, because learning contents themselves may have prerequisites that the user has not mastered yet [31]. The learner-centric adaptation aims at building a personalized learning path based on learner's activity profile. The learner-centric adaptation achieves its functionality in two steps: (i) the system retrieves the related ontology concepts and learning resources by using an elimination process in the following order: similar knowledge, prerequisite knowledge, core knowledge, and related knowledge; (ii) the system removes learning concepts, learning resources, and learning paths that have already been covered by the learner.

Table 4.8 shows some of the rules used to derive an optimum learning path that avoids reiterated covered concepts and consumed learning resources. The property $IsMappedTo(?C, ?LO)$ maps the concept related to the learner's query to a corresponding learning resource. *SimilarLearningResourceRules-1* in Table 4.8 establishes a temporal constraint represented by the temporal relationship *SimilarLR* between the activity ID *(a)* and learning resource *(LR)*. The properties $\neg Covered(?L, ?C_i)$ and $\neg Consumed(?L, ?LO_j)$ relate an individual learner to a concept or a learning resource that has not been covered or consumed so far. It should be noted that the system automatically establishes relations of type $\neg Covered(?L, ?C_i)$ and $\neg Consumed(?L, ?LO_j)$ for all those concepts and resources that have not been covered or consumed by a particular learner. The learning sequence is generated by applying, in order, the following relationships: *Isa*, *HasPrerequisite*, *NecessaryPartOf*, and *Partof*.

SWRL Rules	**SimilarLearningResourceRule-1:** ConductedLearningActivity(?L,?a)∧ MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧IsMappedTo(?C,?LR) →SimilarLR(?a,?LR) **SimilarLearningResourceRule-2:** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧Has(?C,?Ci)∧¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬Consumed(?L,?LRi)→SimilarLR(?a,?LRi) **SimilarLearningResourceRule-3:** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧Isa(?C,?Ci)∧¬ Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬Consumed(?L,?LRi)→SimilarLR(?a,?LRi) **PrerequisiteLearningResourceRule-1:** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧HasPrerequisite(?Q,?Ci)∧¬ Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬Consumed(?L,?LRi) →PrerequisiteLR(?a,?LRi) **CoreLearningResourceRule-1 :** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧HasNecessaryPart(?Q,?Ci)∧¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬ Consumed(?L,?LRi)→CoreLR(?a,?LRi) **CoreLearningResourceRule-2 :** ConductedLearningActivity(?L, ?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧IsNecessaryPartOf(?Q,?Ci)∧¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬Consumed(?L, ?LRi)→CoreLR(?a,?LRi) **NonCoreRelatedLearningResourceRule-1:** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧HasPart(?Q,?Ci)∧¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧¬Consumed(?L,?LRi) →NonCoreRelatedLR(?a,?LRi) **NonCoreRelatedLearningResourceRule-2:** ConductedLearningActivity(?L,?a)∧MakeQuery(?a,?Q)∧ HasKeyword(?Q,?C)∧IsPartOf(?Q,?Ci)∧ ¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧ ¬Consumed(?L,?LRi) →NonCoreRelatedLR(?a,?LRi)

Table 4.8 SWRL Rules for Learning Construction

The relation $Isa(?C,?C_i)$ states that concept C is similar to concept C_i. The relationship $Has\Pr erequisite(?C,?C_i)$ involving concept C and concept C_i, denotes that concept C_i is a prerequisite knowledge of C and needs to be covered prior to it. The core relationship *HasNecessaryPart*$(?C,?C_i)$ represents the necessary part-whole relation where concept C cannot be completely understood without covering concept C_i. In addition, the relationship *HasPart*$(?C,?C_i)$ represents the part-whole relation where concept C_i is part of concept C, in the sense that it represents a related knowledge component of C.

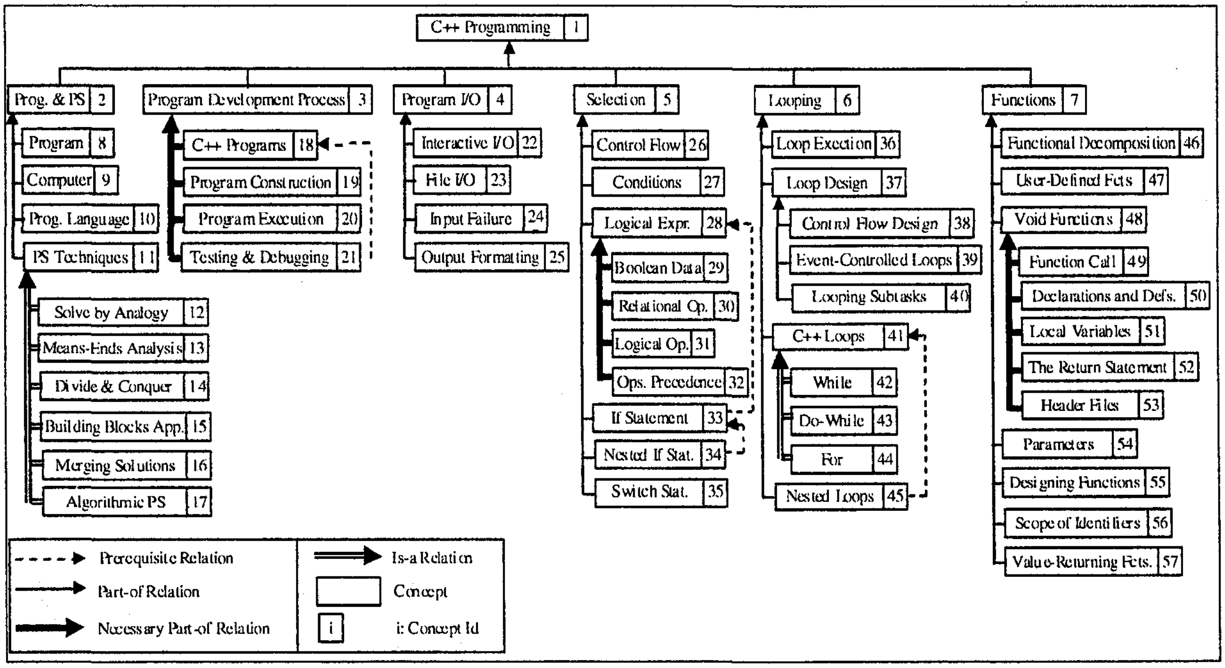


Figure4.5 Domain Ontology for C++ Programming

This section illustrates the logic used for learner adaptation by a learning scenario where learner *Irene* wants to learn about “logic expressions” of the C++ programming language. For this propose, system uses the C++ programming ontology used in the previous chapter and which is reproduced in Figure 4.5 for convenience. For instance, let’s assume that learner *Irene* queries the system with "*Logical Expression*". This query has similar keywords with concept C_{28} which describes *LogicalExpression* in the domain ontology. The reasoning engine is invoked and the system maps learning resources LR_{28a} and LR_{28b} to concept C_{28}. This is represented by facts *A1* in table 4.9. When rule *SimilarLearningResourceRule-1* is applied, facts *A2* is inferred and added to the knowledge base. For this example, it should be noted that concept C_{28} does not have any prerequisite knowledge or similar knowledge in the domain ontology. However, concepts C_{29}, C_{30}, C_{31}, and C_{32} are necessary parts of concept C_{28}. Let's also assume that concepts C_{29} and C_{30} have been covered by *Irene* in previous studies and therefore will not be provided to her at this time. Furthermore, let's assume that Learning resources (LR_{31a}, LR_{31b}) and LR_{32a} correspond to concepts C_{31} and C_{32} respectively, and that learning resources LR_{31b} and LR_{32a} have not been consumed by Irene so far. This info is represented by facts *B1* in Table 4.9. When rule *CoreLearningResourceRule-1* is applied, fact *B2* is inferred and added to the knowledge base. Fact *C1* states that concept C_{28} is part of C_5 and has not been covered by *Irene*. Since LR_{5a} and LR_{5c}, which are the learning resources corresponding to concept C_5, have not been consumed by *Irene* so far; facts in *C2* can be inferred and added to the knowledge base. The above reasoning illustrated by the application of the SWRL rules shown in Table 4.8 produces the learning sequence shown in Figure 4.6.

The learning path shown in Figure 4.6.a is built without considering Irene's previous activity, while the learning path in Figure 4.6.b consists of the optimized learning path using both system-centric adaptations and learner centric adaptations.

	Ontology related Facts	**Inferred Facts**

Facts	***A1)***
	ConductedLearningActivity(Irene,A1)
	MakeQuery(A1, Logical Express)
	HasKeyword(Logical Express, C_{28})
	IsMappedTo(C_{28},LR_{28a}) IsMappedTo(C_{28}, LR_{28b})
		A2)
		SimilarLR
		(A1, LR_{28a})
		SimilarLR
		(A1, LR_{28b})

	B1)
	ConductedLearningActivity(Irene,A1)
	MakeQuery(A1, Logical Express)
	HasKeyword(Logical Express, C_{28})
	HasNecessaryPart(C_{28},C_{29})
	HasNecessaryPart(C_{28},C_{30})
	HasNecessaryPart(C_{28},C_{31})
	HasNecessaryPart(C_{28},C_{32})
	¬Covered(Irene,C_{31})
	¬Covered(Irene,C_{32})
	IsMappedTo(C_{31},LR_{31a})
	IsMappedTo(C_{31},LR_{31b})
	IsMappedTo(C_{32},LR_{32a})
	¬Consumed(Irene,LR_{31b})
	¬Consumed(Irene, LR_{32a})
		B2)
		CoreLR (A1,LR_{31b})
		CoreLR (A1,LR_{32a})

	C1)
	ConductedLearningActivity(Irene,A1)
	MakeQuery(A1, Logical Express)
	HasKeyword(Logical Express, C_{28})
	IsPartOf(C_{28},C_5)
	¬Covered(Irene,C_5)
	IsMappedTo(C_5,LR_{5a})
	IsMappedTo(C_5,LR_{5b})
	IsMappedTo(C_5,LR_{5a})
	¬Consumed(Irene,LR_{5a})
		C2)
		NonCoreRelatedLR
		(A1,LR_{5a})

Table 4.9 Instance for Build Learning Path

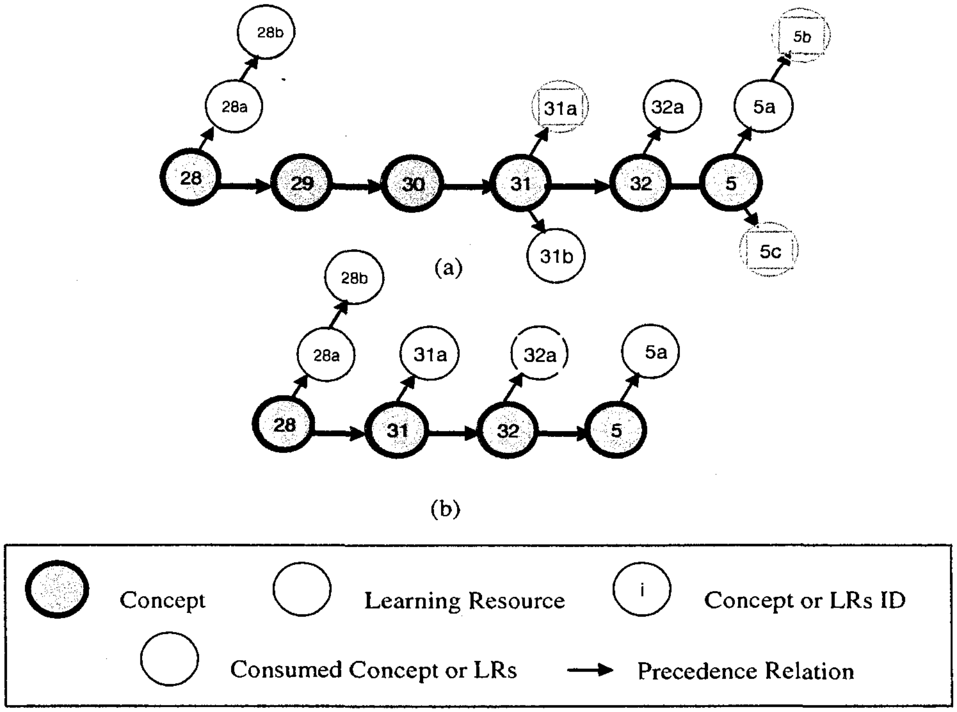
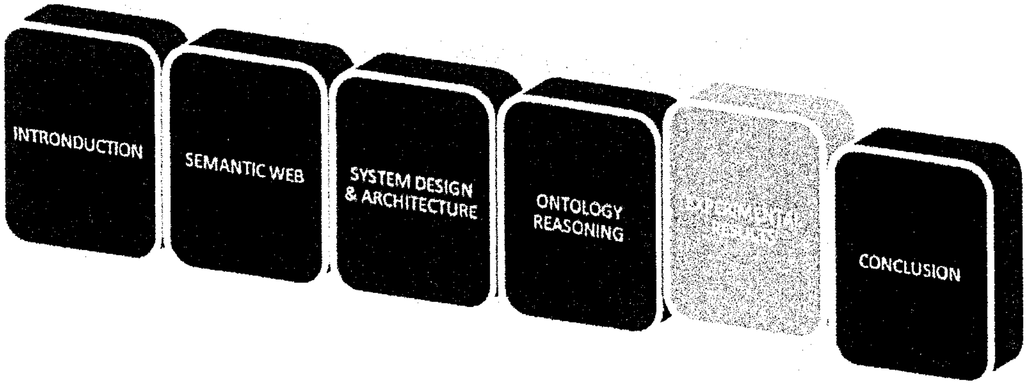


Figure 4.6 Learning Construction

CHAPTER 5

EXPERIMENTAL RESULTS



This section describes the operational environment of the developed system. It then shows some experimental results, and finally, it provides a performance evaluation study. For a better illustration of the system's main functions, this section first describes the ontology authoring and knowledge base construction process, and then, it provides some scenarios to demonstrate the main system services. Finally, it evaluates the performance of the proposed system by comparing it to existing similar systems.

5.1 Ontology and Knowledge Base Construction

To illustrate the ontology authoring process, this section provides an example showing the way we authored the C++ ontology used in chapter 4. It used Protégé to build the global ontology space as shown in Figure 5.1. Figure 5.2 shows the class hierarchy of the global ontology space which consists of learner ontology, activity ontology, device ontology, environment ontology, and domain ontology. Finally, an example ontology describing the subject domain of C++ programming is formally defined in Figure 5.3 using the various properties such *Prerequisite, PartOf, NecessaryPartOf, and Isa.*

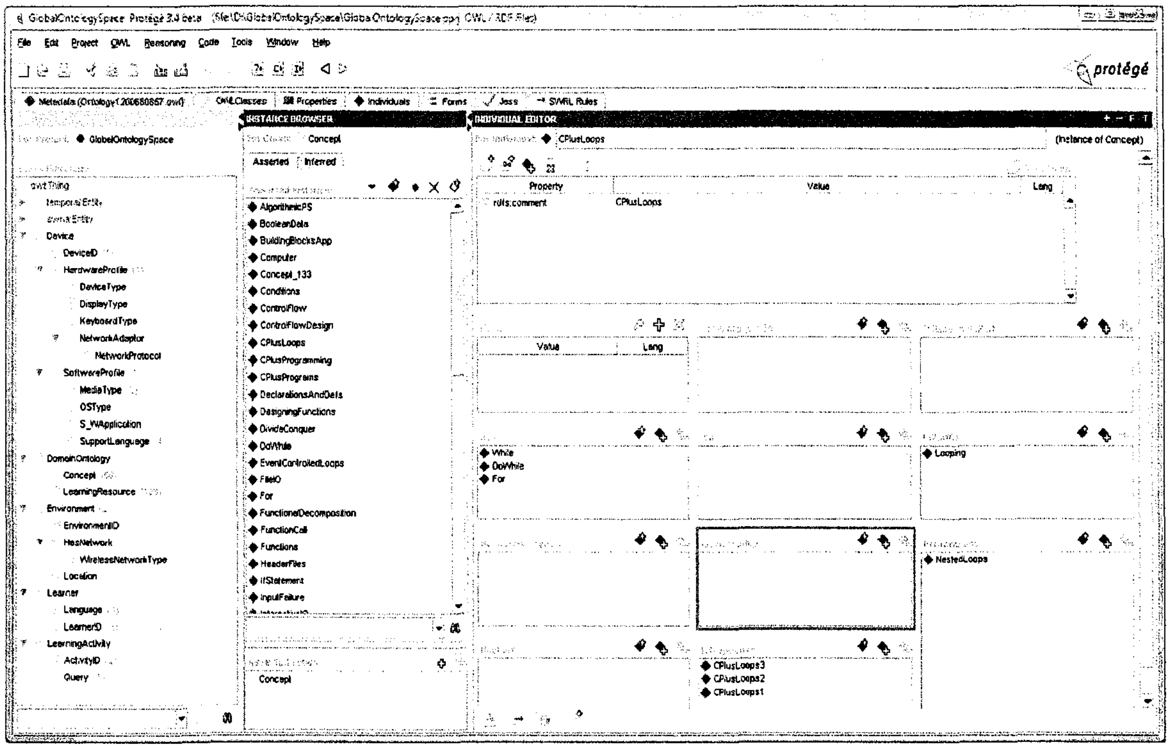


Figure 5.1 Global Ontology Space editing with Protégé

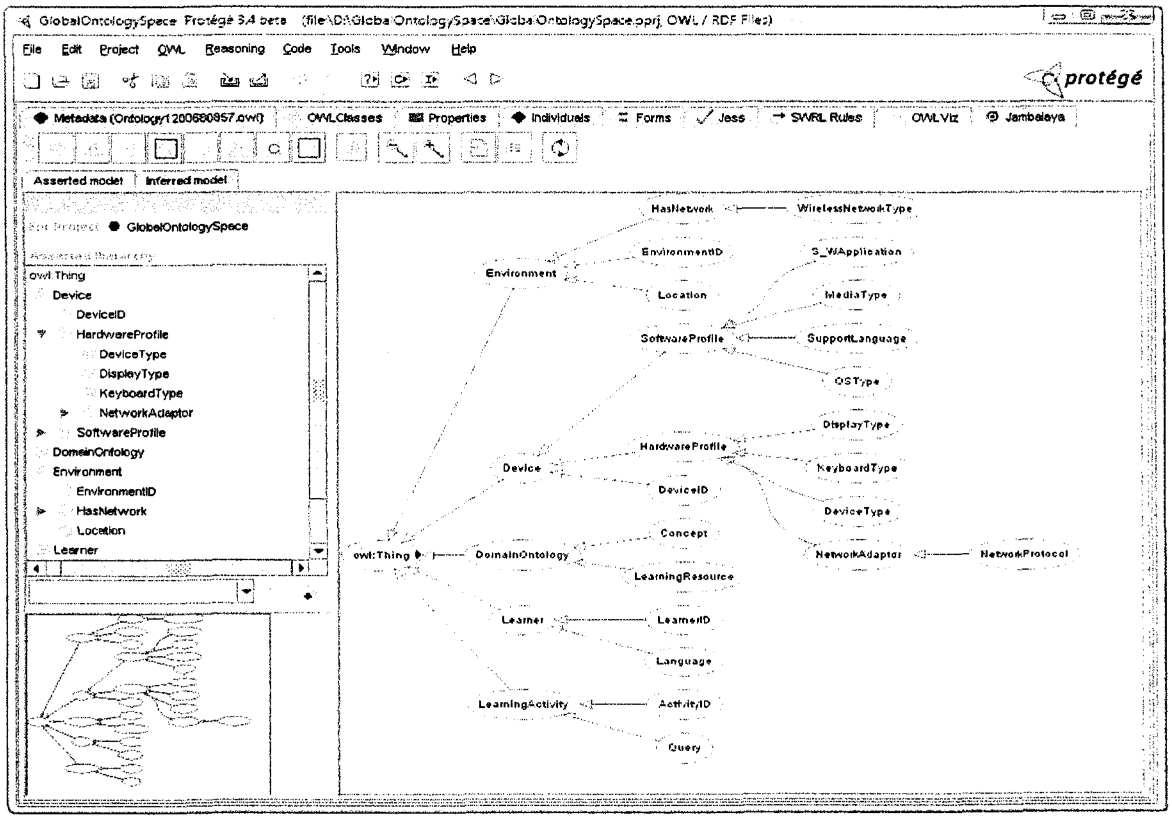


Figure 5.2 Class Hierarchy of the Global Ontology Space

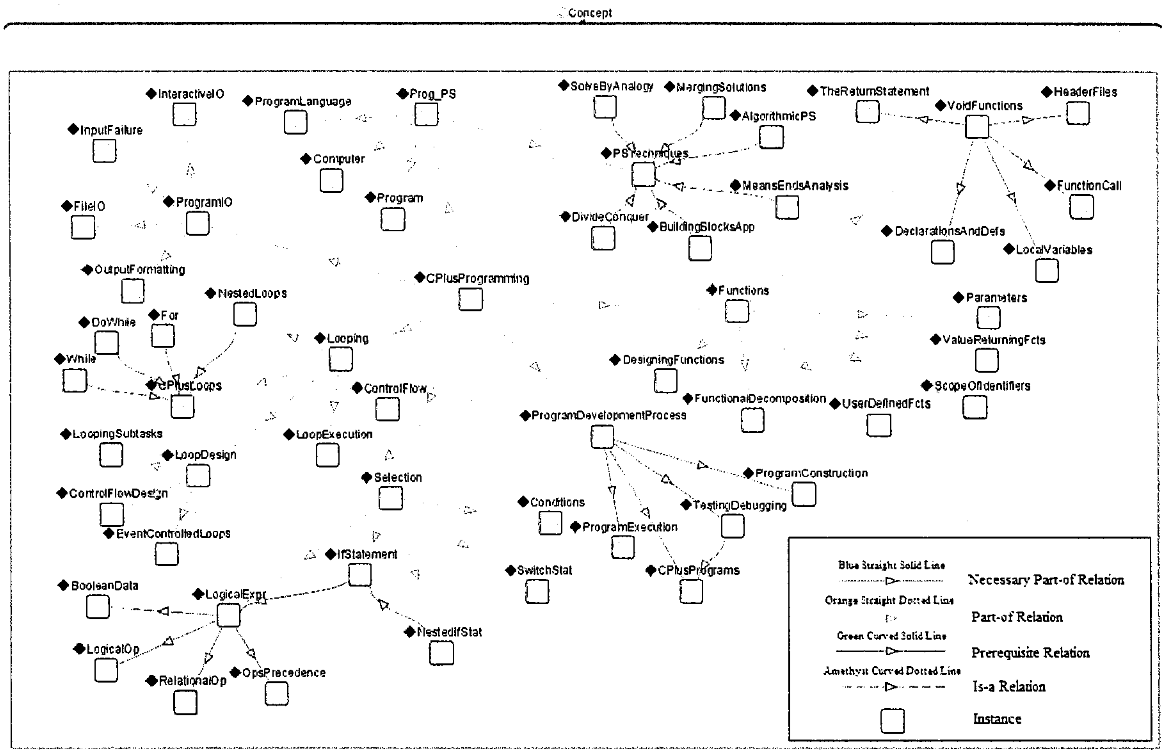
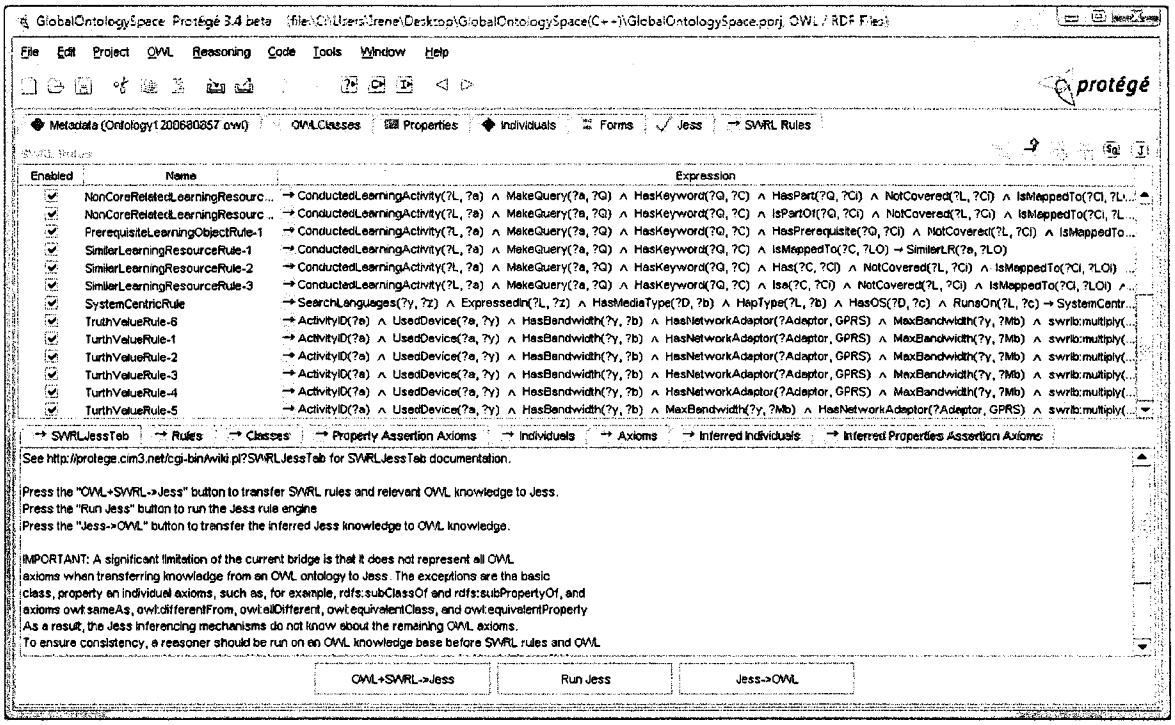


Figure 5.3 Subject Domain Ontology: C++ Programming

SWRL Tab of Protégé (see Figure 5.4) is used to construct the SWRL rules which represent the core of our knowledge base. All SWRL rules have been introduced in chapter 4. These can be classified into system-centric rules (i.e. media rules, language rule) and learner-centric rules (i.e. learning sequence rules). The system-centric rules use the device and environment atomic context elements to infer metadata that can be used to filter out learning resources that are compatible with the system operating environment. The learner-centric rules are used to build a personalized learning path by removing already covered learning concepts, learning resources, and learning sub-path(s). Thus, the ultimate goal of the system is to optimize the learning path using both system-centric adaptations and learner-centric adaptations.


GlobalOntologySpace Protégé 3.4 beta (file:\C:\Users\Irene\Desktop\GlobalOntologySpace(C++)\GlobalOntologySpace.pprj, OWL / RDF Files)
File Edit Project OWL Reasoning Code Tools Window Help
protégé
Metadata (Ontology1200680857.owl)
OWLClasses
Properties
Individuals
Forms
Jess
SWRL Rules
SWRL Rules
SWRLJessTab
Rules
Classes
Property Assertion Axioms
Individuals
Axioms
Inferred Individuals
Inferred Properties Assertion Axioms
OWL+SWRL->Jess
Run Jess
Jess->OWL

Enabled	Name	Expression
☑	NonCoreRelatedLearningResourc...	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ HasPart(?Q, ?Ci) ∧ NotCovered(?L, ?Ci) ∧ IsMappedTo(?Ci, ?L...
☑	NonCoreRelatedLearningResourc...	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ IsPartOf(?Q, ?Ci) ∧ NotCovered(?L, ?Ci) ∧ IsMappedTo(?Ci, ?L...
☑	PrerequisiteLearningObjectRule-1	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ HasPrerequisite(?Q, ?Ci) ∧ NotCovered(?L, ?Ci) ∧ IsMappedTo...
☑	SimilarLearningResourceRule-1	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ IsMappedTo(?C, ?LO) → SimilarLR(?a, ?LO)
☑	SimilarLearningResourceRule-2	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ Has(?C, ?Ci) ∧ NotCovered(?L, ?Ci) ∧ IsMappedTo(?Ci, ?LOi) ...
☑	SimilarLearningResourceRule-3	→ ConductedLearningActivity(?L, ?a) ∧ MakeQuery(?a, ?Q) ∧ HasKeyword(?Q, ?C) ∧ Isa(?C, ?Ci) ∧ NotCovered(?L, ?Ci) ∧ IsMappedTo(?Ci, ?LOi) ∧...
☑	SystemCentricRule	→ SearchLanguages(?y, ?z) ∧ ExpressedIn(?L, ?z) ∧ HasMediaType(?D, ?b) ∧ HapType(?L, ?b) ∧ HasOS(?D, ?c) ∧ RunsOn(?L, ?c) → SystemCentr...
☑	TruthValueRule-6	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?Adaptor, GPRS) ∧ MaxBandwidth(?y, ?Mb) ∧ swrlb:multiply(...
☑	TurthValueRule-1	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?Adaptor, GPRS) ∧ MaxBandwidth(?y, ?Mb) ∧ swrlb:multiply(...
☑	TurthValueRule-2	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?Adaptor, GPRS) ∧ MaxBandwidth(?y, ?Mb) ∧ swrlb:multiply(...
☑	TurthValueRule-3	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HesNetworkAdaptor(?Adaptor, GPRS) ∧ MaxBandwidth(?y, ?Mb) ∧ swrlb:multiply(...
☑	TurthValueRule-4	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ HasNetworkAdaptor(?Adaptor, GPRS) ∧ MaxBandwidth(?y, ?Mb) ∧ swrlb:multiply(...
☑	TurthValueRule-5	→ ActivityID(?a) ∧ UsedDevice(?a, ?y) ∧ HasBandwidth(?y, ?b) ∧ MaxBandwidth(?y, ?Mb) ∧ HasNetworkAdaptor(?Adaptor, GPRS) ∧ swrlb:multiply(...

```
See http://protege.cim3.net/cgi-bin/wiki.pl?SWRLJessTab for SWRLJessTab documentation.

Press the "OWL+SWRL->Jess" button to transfer SWRL rules and relevant OWL knowledge to Jess.
Press the "Run Jess" button to run the Jess rule engine
Press the "Jess->OWL" button to transfer the inferred Jess knowledge to OWL knowledge.

IMPORTANT: A significant limitation of the current bridge is that it does not represent all OWL
axioms when transferring knowledge from an OWL ontology to Jess. The exceptions are the basic
class, property en individual axioms, such as, for example, rdfs:subClassOf and rdfs:subPropertyOf, and
axioms owl:sameAs, owl:differentFrom, owl:allDifferent, owl:equivalentClass, and owl:equivalentProperty
As a result, the Jess inferencing mechanisms do not know about the remaining OWL axioms.
To ensure consistency, a reasoner should be run on an OWL knowledge base before SWRL rules and OWL
```

Figure 5.4 Overview of Knowledge Base Construction using SWRL Tab of Protégé

5.2 Experimental Results

To illustrate the ontology reasoning mechanisms used in this study, this section provides a number of scenarios to demonstrate the various system-centric adaptations and learner-centric adaptations. For system-centric adaptations, system used the *C++ programming language* ontology as described in Figure 5.3. The system asked few learners to make queries related to *C++ programming language* using devices with different software and hardware capabilities. Figure 5.5 shows the used mobile phones, while Table 5.1 shows their capabilities and surrounding environment. In particular, for the current bandwidth system used the predicted network bandwidth at the time we performed the experiment.

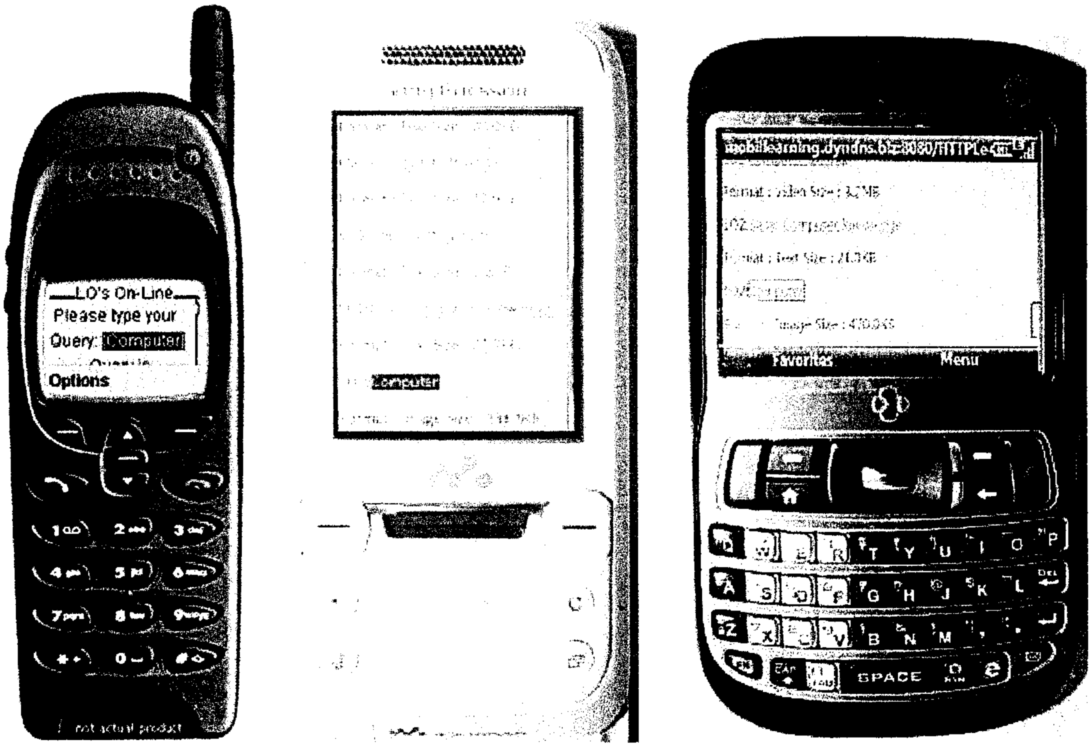


Figure 5.5 Experimental Results with Different Mobile Phone

	Basic Nokia phone emulator	Sony Ericsson W830C	HTC S621
Operating system	Symbian	Sony Ericsson Java	Windows Mobile 6 Standard
Available Memory	256.0kbytes	6.0Mbytes	32.0Mbytes
Connection Speed	32.0kbps	48.0kbps	120.0kbps
Current Bandwidth	8.0kbps	16.0kbps	80.0kbps
Screen Resolution	128*96 pixels	320*240 pixels	320*240 pixels
Support Language(s)	English	English, French…	English, French, Chinese…
Keyboard Type	Virtual	Virtual	Real
Media Type	Text, Image	Text, Image, Video	Text, Image, Video
Display Type	Monochrome	256k Colors	65536 Colors
Network Adaptor	GSM 1900	GPRS, EDGE	Wifi, Bluetooth, EDGE
Browser	WAP 2.0	WAP 2.0	Internet Browser (WWW)

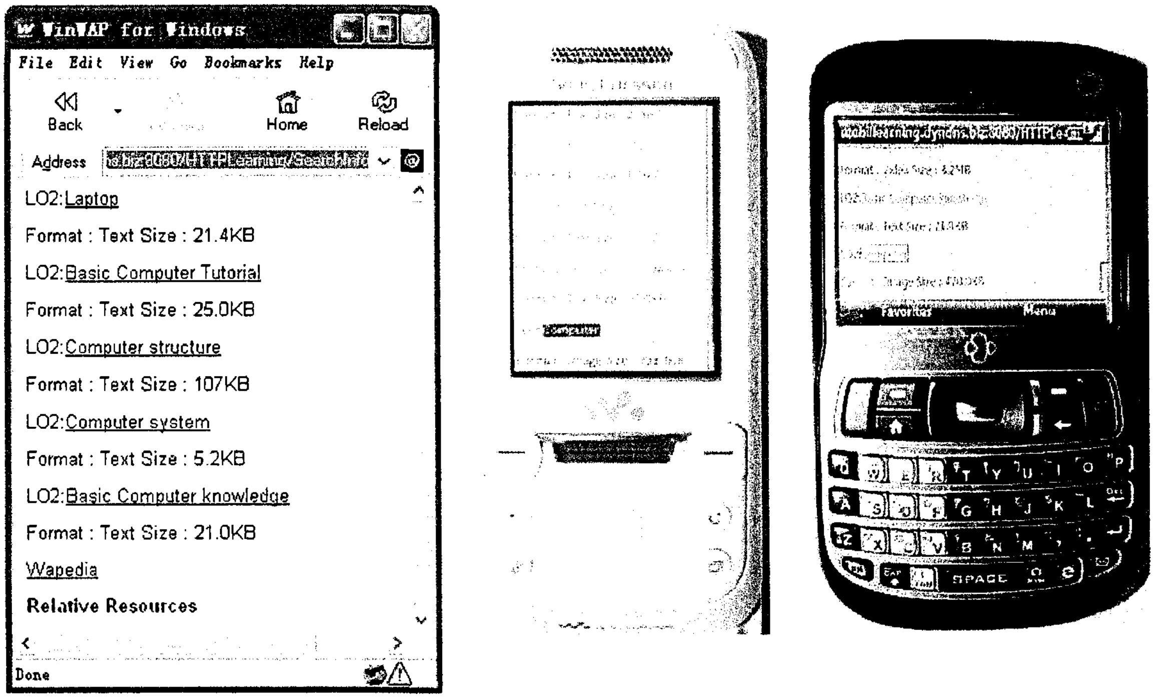
Table 5.1 Software & Hardware Capabilities for Mobile Phone

Below is a fragment of the OWL description for the mobile phone showing the capabilities of HTC S621.

```
<UsedDevice rdf:ID="HTCS621">
   <HasSupportLanguage rdf:resource="#English"/>
   <HasSupportLanguage rdf:resource="#French"/>
   <HasSupportLanguage rdf:resource="#Chinese"/>
   <HasDisplayType> <DisplayType rdf:ID="65536Colors"/> </HasDisplayType>
   <HasOS rdf:resource="#WindowsMobile6Standard"/>
   <AvailableMemory rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
     32.0
   </AvailableMemory>
   <RunApplication> <S_WApplication rdf:ID="Image"/> </RunApplication>
   <RunApplication> <S_WApplication rdf:ID="Text"/> </RunApplication>
   <RunApplication> <S_WApplication rdf:ID="Video"/> </RunApplication>
   <HasKeyboardType> <KeyboardType rdf:ID="Real"/> </HasKeyboardType>
   <HasNetworkAdaptor> <NetworkAdaptor rdf:ID="Wifi"/> </HasNetworkAdaptor>
   <HasNetworkAdaptor> <NetworkAdaptor rdf:ID="Bluetooth"/> </HasNetworkAdaptor>
   <HasNetworkAdaptor> <NetworkAdaptor rdf:ID="EDGE"/> </HasNetworkAdaptor>
   <MaxConnectionSpeed rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
     120.0
   </MaxConnectionSpeed>
   <HasBandwidth rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
     80.0
   </HasBandwidth>
   <HasScreenWidth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">240</HasScreenWidth>
   <HasScreenLength rdf:datatype="http://www.w3.org/2001/XMLSchema#int">320</HasScreenLength>
</UsedDevice>
```

5.2.1 System-centric adaptation

Section 4.2 described the method used to filter out learning resources using system-centric adaptations based on device and environment context elements. This section presents some experimental results related to the system-centric adaptations obtained using the three mobile phones shown in Table 5.1. In particular, the following steps have been adopted to achieve these experimental results. The system is programmed to sense the actual network bandwidth from time to time, and whenever queries are made, the current bandwidth is predicted as shown in section 4.2, and then translated into meaningful symbolic values such as low, medium, and high bandwidth. For example, the bandwidth predicted by the system for the HTC S621 was "high bandwidth". Based on this assumption, the system calculates the maximum data file size and selects the media type of retrieved learning resources to ensure that they can be played on the device. This step is achieved by applying *Truth Value Rules, File Size Rule, Media type Rules*, and *Allowed File Size Rules* respectively as shown in section 4.2. When these rules are applied, a number of facts are inferred resulting in the addition of many statements to the list of facts. The next processing step is to select the language of retrieved learning resources to make sure that this is similar to the learner's preferred language, and that it can be supported by the mobile device. This is achieved by applying *LanguageRule-1* as given in section 4.2. Once the language rule is applied, a new fact is inferred resulting in the addition of statement *SearchLanguage(UserName, Language)* to the list of facts. It should be noted here that there might be some resources that require software applications that are not supported by the used mobile device. So, the final step is to check whether the software applications required to play the retrieved learning resources are supported by the used mobile device. This is achieved by applying *SystemCentricRule-1* given in section 4.2. Figure 5.6 describes the results of applying these rules on the three used mobile devices as shown in Figure 5.5. It can be seen that the Nokia phone emulator has low bandwidth. The Nokia emulator was used because it was not possible to experiment our system on a real old cell phone device. Therefore, system has tested the system-centric adaptations on the Nokia phone emulator which obviously has limited hardware and software features. Due to the limited resources, the learning resources retrieved for the Nokia phone have been reduced to text media type not exceeding 125Kbytes as shown in Figure 5.6.a. The same experimental results for the Sony Ericsson W830C are presented in Figure5.6.b. These include some learning resources of image media type in addition to text media type and were restricted to a maximum size of 250Kbytes. Figure 5.6.c however, shows the retrieved learning resources for the HTC S621 smart phone. This device can support all types of media without any size restriction. This experiment assumes that English is the preferred language for the learners.



LO2:Laptop

Format : Text Size : 21.4KB

LO2:Basic Computer Tutorial

Format : Text Size : 25.0KB

LO2:Computer structure

Format : Text Size : 107KB

LO2:Computer system

Format : Text Size : 5.2KB

LO2:Basic Computer knowledge

Format : Text Size : 21.0KB

Wapedia

Relative Resources

(a) (b) (c)

Figure 5.6 System-Centric Adaptations for Various Devices

5.2.2 Learner-Centric Adaptation

Chapter 4 showed the method used to generate personalized learning paths tailored to the needs of the learners. This section presents some experimental results that illustrate the various learner-centric adaptations. Let's assume *Irene* is working on an assignment for comparing some "object oriented programming languages". Being a Java programmer with a little knowledge about C++, *Irene* is confused about the syntax of "C++ Loops".

So, she used her mobile device to query the system using "C++ Loops". Let's also assume that she has previously used our system to query other concepts. Once logged in, the system tracks her previous login sessions, covered concepts, consumed learning resources, and previously conducted learning interactions. The system will then proceed with the following steps which describes the whole learning scenario.

Step1: the system uses the keywords in the learner' query and accesses the related subject-domain ontology to infer those concepts that are part of similar knowledge, prerequisite knowledge, core-knowledge, and related knowledge using *Is-a*, *HasPrerequisite, HasNecessaryPart,* and *HasPart* properties respectively. Consequently, the concepts *While, Do-While,* and *For* are inferred and classified as "similar knowledge" to *C++ Loops*. Similarly, concept *Looping* is inferred and classified as "related knowledge" to *C++ Loops*. As shown in the C++ ontology no prerequisite or core sub-concepts are allocated to concepts *C++ Loops*. So, in the next processing stage, the system searches for the learning resources associated with the previously inferred ontology concepts by using the following elimination order: similar knowledge, perquisite knowledge, core knowledge, and then related knowledge if any. For our case, learning resources LO_{41a} and LO_{41b} correspond to concept C_{41}*(C++Loops)*. The learning sequence of this scenario is C_{41} *(C++ Loops)* → C_{42} *(While)* → C_{43} *(Do-While)* → C_{44} *(For)* → C_6 *(Looping)*. In the third stage, the system builds a personalized learning path by removing already covered concept, learning resources, and learning sub-path(s). This step is achieved by applying the learning sequence rules as shown in the previous chapter. Once the learning sequence rules are applied, inferred facts are added to the knowledge base. Figure 5.7.a shows the learning path generated by applying the learning sequence rules without taking into consideration *Irene's* previous knowledge. The corresponding experimental results using the WinWap emulator smart phone are presented in Figure 5.8. Figure 5.7.b shows the generated personalized learning path after removing the already covered concepts, learning resources, and learning sub-path(s). The experimental results for this learning path are presented in Figure 5.9.

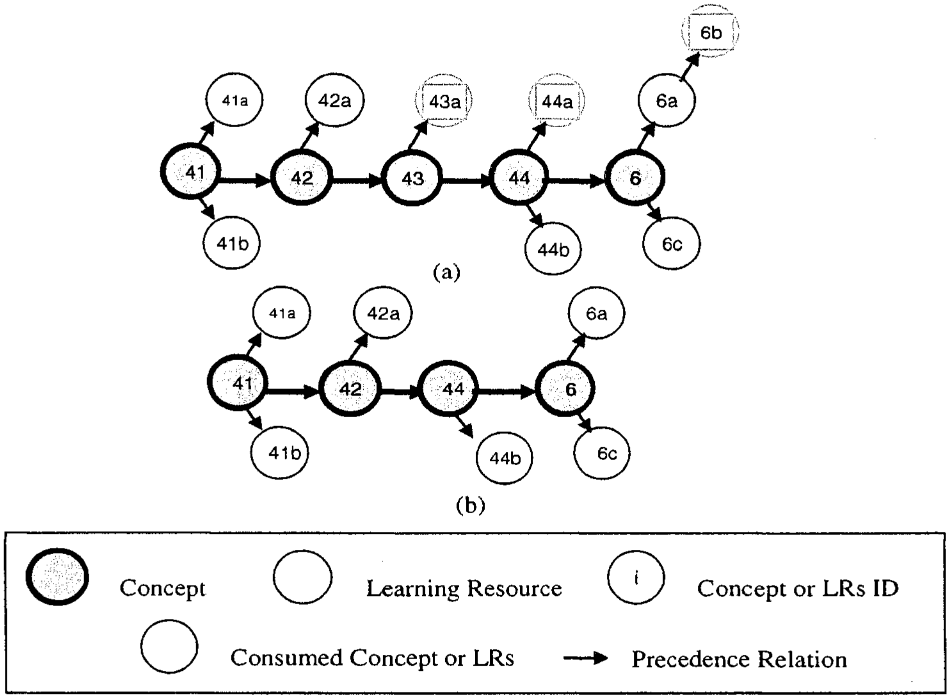
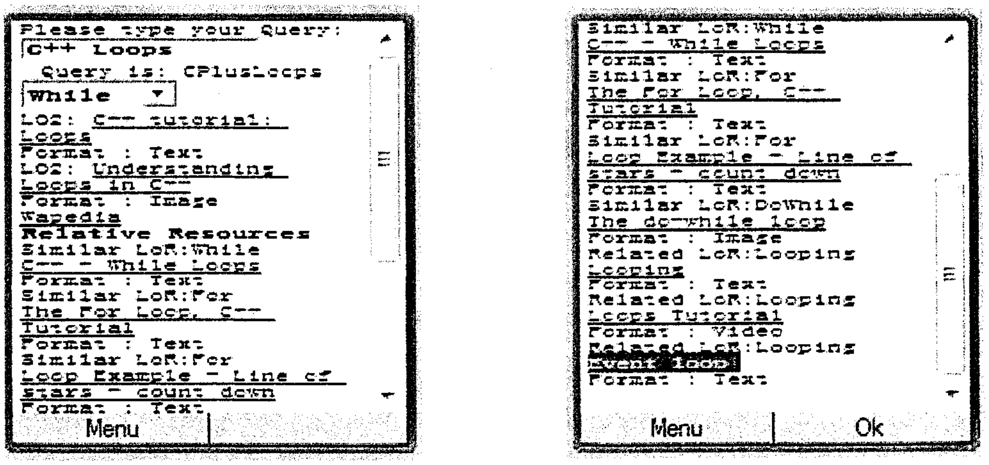


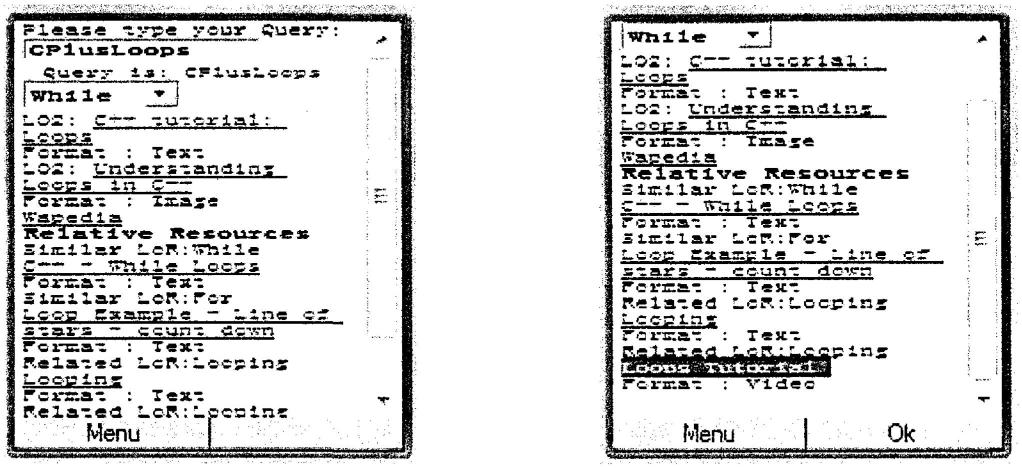
Figure 5.7 Learning Path for Concept C_{41}



```
Please type your Query:
C++ Loops
 Query is: CPlusLoops
While
LO2: C++ tutorial:
Loops
Format : Text
LO2: Understanding
Loops in C++
Format : Image
Wapedia
Relative Resources
Similar LoR:While
C++ - While Loops
Format : Text
Similar LoR:For
The For Loop, C++
Tutorial
Format : Text
Similar LoR:For
Loop Example - Line of
stars - count down
Format : Text
```

```
Similar LoR:While
C++ - While Loops
Format : Text
Similar LoR:For
The For Loop, C++
Tutorial
Format : Text
Similar LoR:For
Loop Example - Line of
stars - count down
Format : Text
Similar LoR:DoWhile
The do-while loop
Format : Image
Related LoR:Looping
Looping
Format : Text
Related LoR:Looping
Loops Tutorial
Format : Video
Related LoR:Looping
Event loop
Format : Text
```

Figure 5.8 Example of Learning Path without Consider Learner's Previous Activity

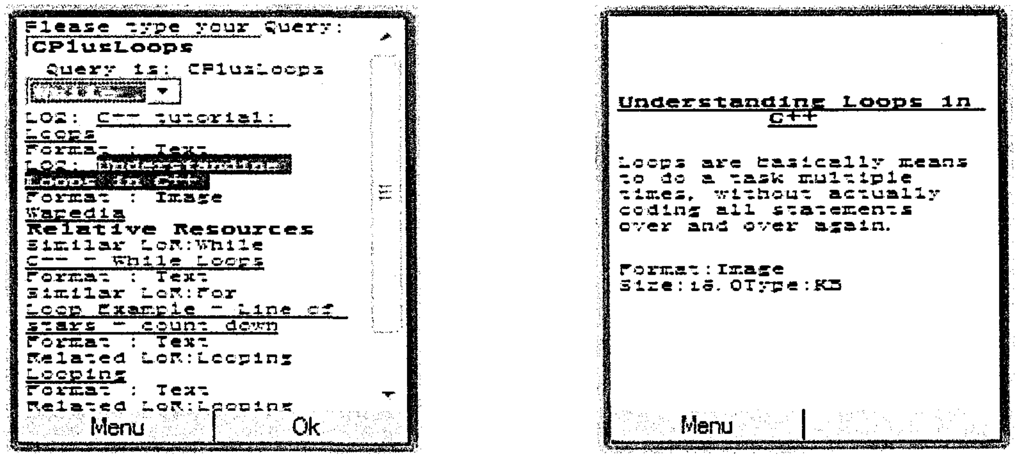


```
Please type your Query:
CPlusLoops
  Query is: CPlusLoops
While
LO2: C++ tutorial:
Loops
Format : Text
LO2: Understanding
Loops in C++
Format : Image
Wapedia
Relative Resources
Similar LoR:While
C++ - While Loops
Format : Text
Similar LoR:For
Loop Example - Line of
stars - count down
Format : Text
Related LoR:Looping
Looping
Format : Text
Related LoR:Looping
Menu
```

```
While
LO2: C++ tutorial:
Loops
Format : Text
LO2: Understanding
Loops in C++
Format : Image
Wapedia
Relative Resources
Similar LoR:While
C++ - While Loops
Format : Text
Similar LoR:For
Loop Example - Line of
stars - count down
Format : Text
Related LoR:Looping
Looping
Format : Text
Related LoR:Looping
Loops Tutorial
Format : Video
Menu | Ok
```

Figure 5.9 Example of Learning Path Recommended to the Learner

The figure below shows the interaction with the learner after presenting the previously recommended learning path. Figure 5.10.b shows the details of the learning resource after being chosen as highlighted in Figure 5.10.a. The system will then automatically add the newly explored learning resource to the list of consumed learning resources.



```
Please type your Query:
CPlusLoops
  Query is: CPlusLoops
LO2: C++ tutorial:
Loops
Format : Text
LO2: Understanding
Loops in C++
Format : Image
Wapedia
Relative Resources
Similar LoR:While
C++ - While Loops
Format : Text
Similar LoR:For
Loop Example - Line of
stars - count down
Format : Text
Related LoR:Looping
Looping
Format : Text
Related LoR:Looping
Menu | Ok
```

(a)

```
Understanding Loops in
C++

Loops are basically means
to do a task multiple
times, without actually
coding all statements
over and over again.

Format:Image
Size:16.0Type:KB
Menu
```

(b)

Figure 5.10 Example of Select Learning Resource

In general, when the learner selects a new concept or sends a new query, learner's expertise is automatically updated and the personalized learning path is re-adjusted by resuming processing steps 4 to 9 as shown in the algorithm given in Figure 4.1.

To show the various logical steps that can be adopted to personalize the learning path, this section provides another scenario. Now, let's assume that *Irene* selects the concept "Looping" which is submitted as a new query to the system. Based on C++ programming ontology, the concepts *Loop Execution, Loop Design, C++ Loops, Nested Loops,* and *C++Programming* are inferred as "related knowledge" of the query *Looping.* The concept *C++ Loops* is inferred as "prerequisite knowledge" of the concept *Nested Loops.* Therefore, the system will suggest the following sequence in ordering the concepts to be provided to the learner (i.e. C_6 *(Looping)* → C_{36} *(Loop Execution)* → C_{37} *(Loop Design)* → C_{41} *(C++ Loops)* → C_{45} *(Nested Loops)),* and their associated learning resources are searched for and retrieved as shown in Figure 5.11.a. Figure 5.11.b shows a personalized learning path based on already removed covered concepts, learning resources, and learning sub-path(s). For example, learning resource LR_{41b} *Understanding Loops in C++* which has already been consumed is therefore removed from the learning path as shown in 5.11.b. Figure 5.12 shows some of the learning resources recommended to *Irene* as a result of her query about the concept "Looping".

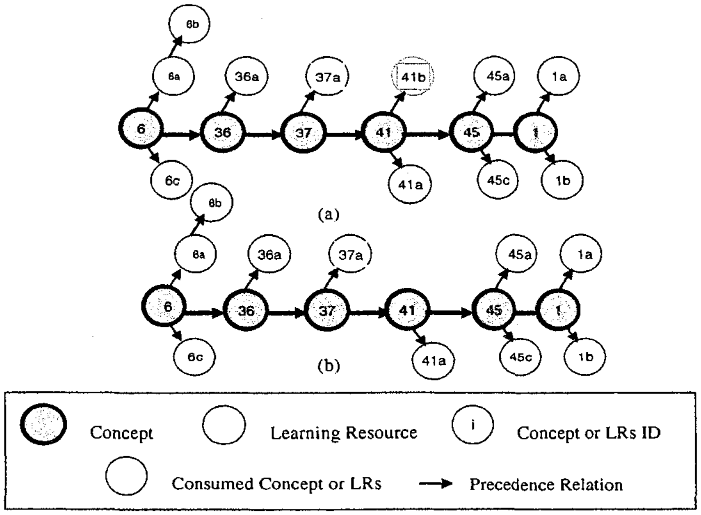
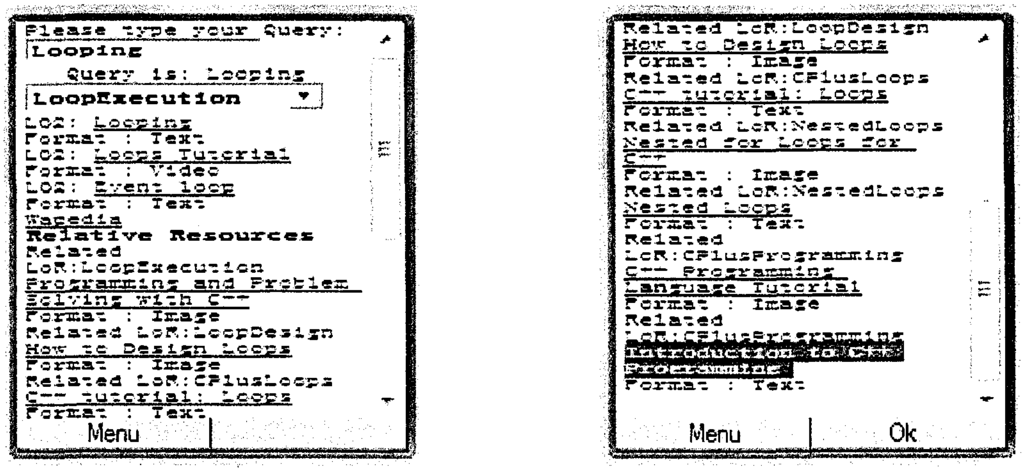


Figure 5.11 Learning Path for Concept C_6



(a)

```
Please type your Query:
Looping
   Query is: Looping
LoopExecution
LO2: Looping
Format : Text
LO2: Loops Tutorial
Format : Video
LO2: Event loop
Format : Text
Wapedia
Relative Resources
Related
LoR:LoopExecution
Programming and Problem
Solving with C++
Format : Image
Related LoR:LoopDesign
How to Design Loops
Format : Image
Related LoR:CPlusLoops
C++ tutorial: Loops
Format : Text
```

(b)

```
Related LoR:LoopDesign
How to Design Loops
Format : Image
Related LoR:CPlusLoops
C++ tutorial: Loops
Format : Text
Related LoR:NestedLoops
Nested for Loops for
C++
Format : Image
Related LoR:NestedLoops
Nested Loops
Format : Text
Related
LoR:CPlusProgramming
C++ Programming
Language Tutorial
Format : Image
Related
LoR:CPlusProgramming
Introduction to C++
Programming
Format : Text
```

Figure 5.12 Example of Learning Path

Recommended to the Learner

5.2.3 Experimental results with photography ontology

This section tries our system using another domain ontology. It uses the same global ontology space by substituting the C++ ontology with the photography ontology which is formally defined in Figure 5.13. The photography ontology is also described using properties *necessary-part-of, part-of, prerequisite, and is-a*. This scenario assumes that *Irene* would like to purchase a camera for her friend as a gift. In order to have a background about this domain to be able to make the right purchase decision, she used her mobile device *Sony Ericsson W830C* to query the system using *Camera* as a keyword. Figure 5.14 illustrates *Irene*'s profile while Figure 5.15 shows the *Sony Ericsson W830C*'s software and hardware capabilities.

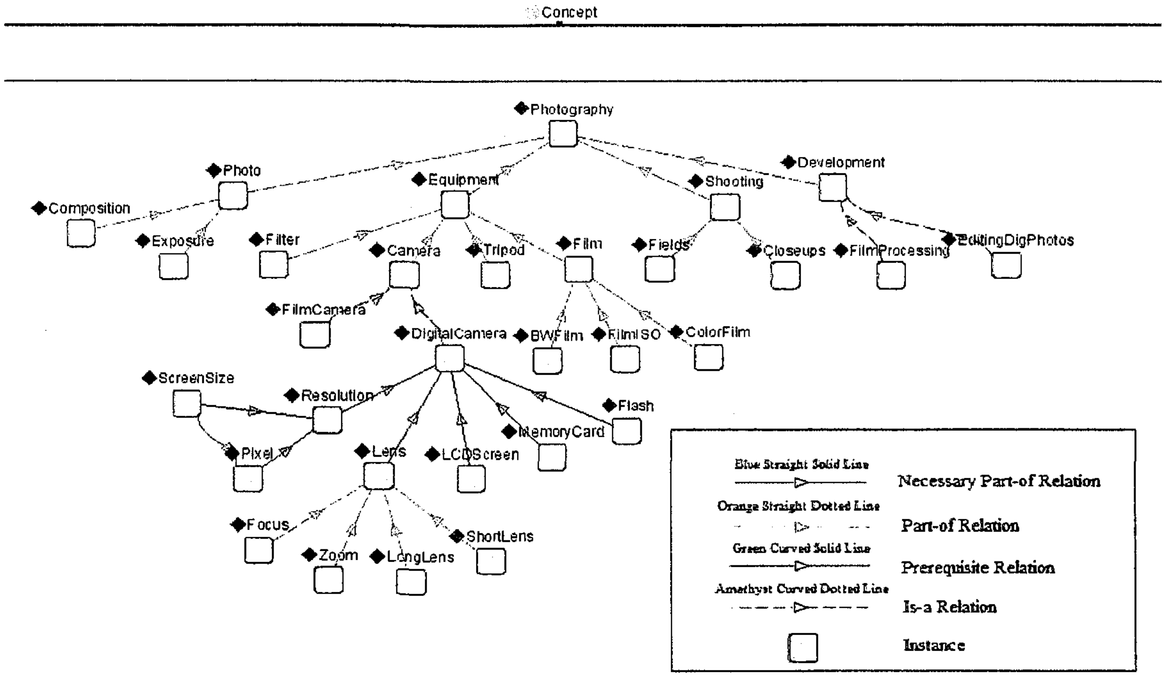


Figure 5.13 Photography Ontology

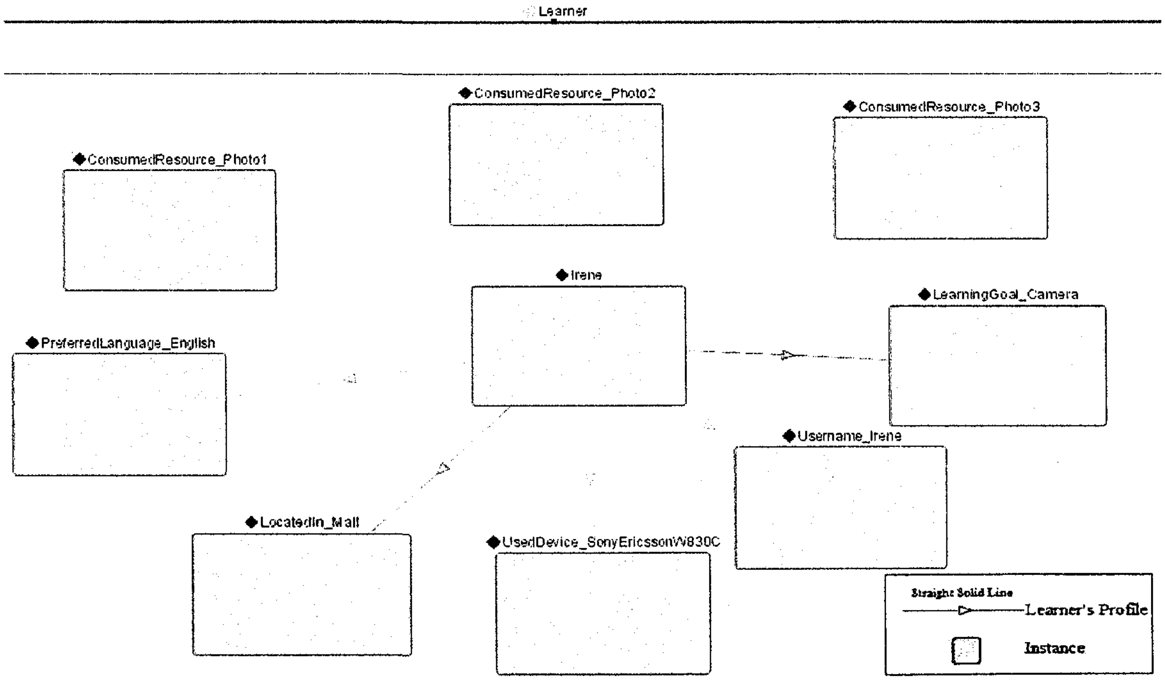


Figure 5.14 Learner's Profile

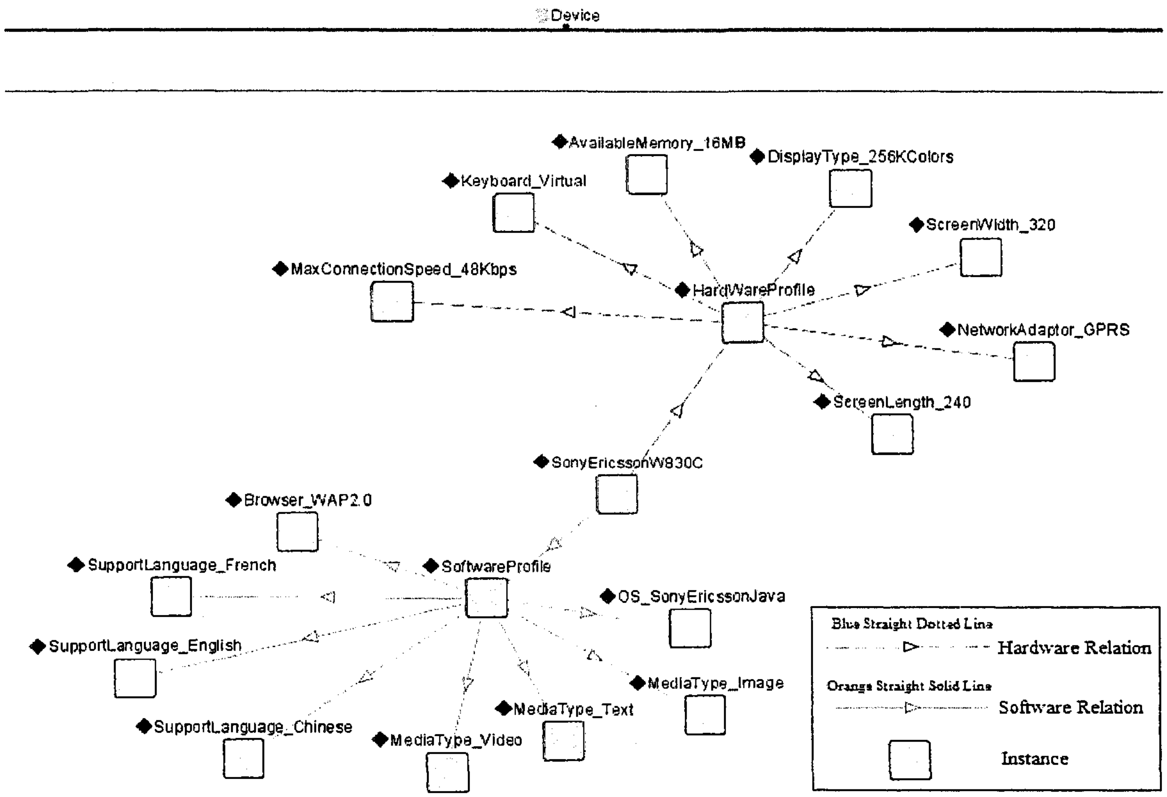
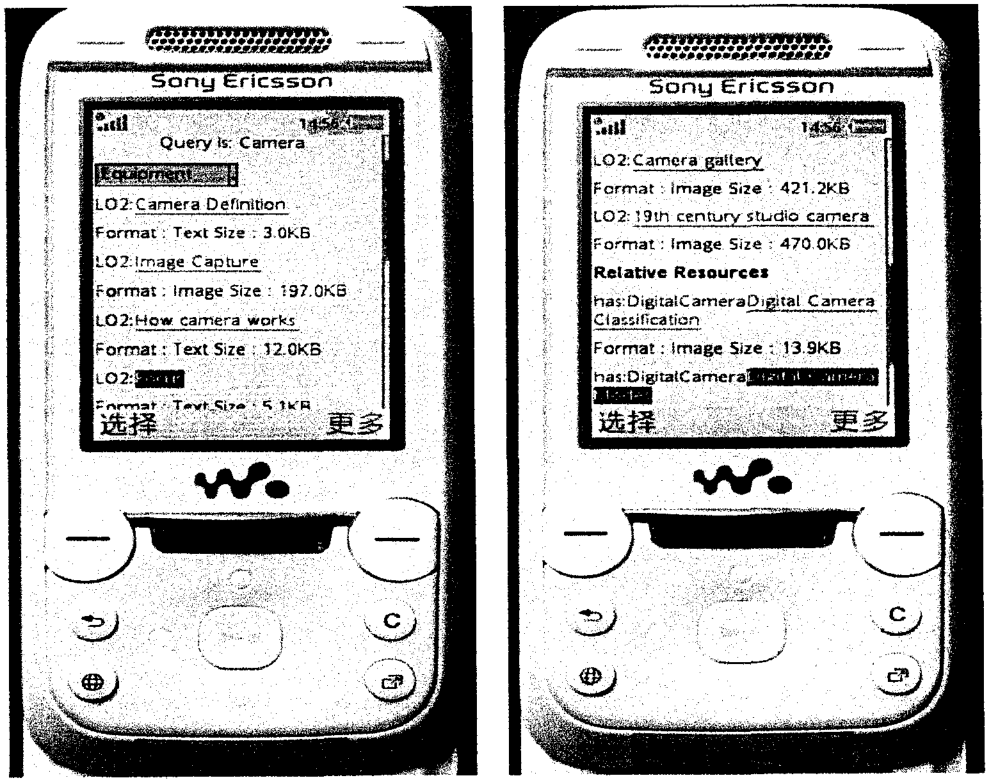


Figure 5.15 Software & Hardware Capabilities for Sony EricssonW830

Let's assume that *Irene* is using her device *Sony Ericsson W830C* with an actual network bandwidth of 40.0kbps. For system-centric adaptation, when *Truth Value Rules* are applied, new facts are inferred resulting in the addition of some statements to the list of facts. These new facts reveal the probabilities for the predicted current bandwidth which was found to be 0.084 for medium bandwidth and 0.916 for high bandwidth. Then, the system uses the obtained values to predict the maximum data file size and to select the appropriate media type for the used device. When *Media Type Rules* and *Allowed File Size Rule* are applied, facts *HasMediaType(SonyEricssonW830C, Text/Image/Video)* and *AllowedSize(SonyEricsson W830C, 546.875)* are inferred, resulting in the addition of other statements to the list of facts. The system infers that the learning resources to be retrieved for the *Sony Ericsson W830C* device should be of text, image, or video media types and should not exceed 546.875Kbytes. The system will finally perform few processing step for system-centric adaptation to check whether those retrieved learning resources can be played on the used device. For learner-centric adaptations however, the system retrieves the related ontology concepts and learning resources by using the following elimination order: similar knowledge, prerequisite knowledge, core knowledge, and related knowledge. In our experiment, the concept *Digital Camera* and *Film Camera* are inferred as "similar knowledge" of the concept *Camera*. The concept *Equipment* is inferred as "related knowledge" of concept *Camera*. The learning sequence for this scenario is $C_9(Camera) \rightarrow C_{16}(DigitalCamera) \rightarrow C_{17}(FilmCamera) \rightarrow C_3(Equipment)$. When *Learning Resources Rules* are applied, the system builds a personalized learning path by removing already covered concepts, learning resources, and learning sub-path(s). Figure 5.16 shows the experimental results for this scenario. All retrieved learning resources are of text, image, or video media type. The data file size is smaller than 546.875KB. Figure 5.17 shows an instance of learning resource proposed to the learner.



Query Is: Camera

Equipment

LO2: Camera Definition

Format : Text Size : 3.0KB

LO2: Image Capture

Format : Image Size : 197.0KB

LO2: How camera works

Format : Text Size : 12.0KB

LO2: [illegible]

Format : Text Size : 5.1KB

选择 更多

LO2: Camera gallery

Format : Image Size : 421.2KB

LO2: 19th century studio camera

Format : Image Size : 470.0KB

Relative Resources

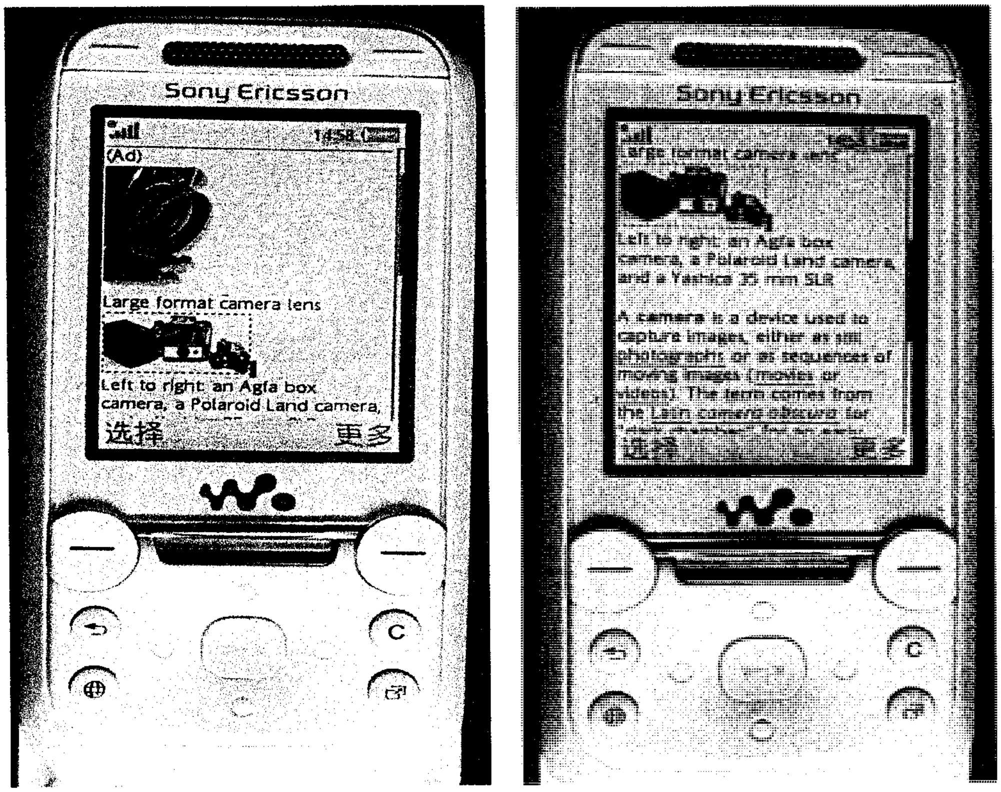
has:DigitalCameraDigital Camera Classification

Format : Image Size : 13.9KB

has:DigitalCamera[illegible]

选择 更多

Figure 5.16 Retrieved Learning Resources



(Ad)

Large format camera lens

Left to right: an Agfa box camera, a Polaroid Land camera,

选择 更多

Large format camera lens

Left to right: an Agfa box camera, a Polaroid Land camera, and a Yashica 35 mm SLR

A camera is a device used to capture images, either as still photographs or as sequences of moving images (movies or videos). The term comes from the Latin camera obscura for

选择 更多

Figure 5.17 Individual of Learning Resource

5.3 System Evaluation

This section compares the proposed mobile learning system with two similar systems – context-aware E-learning [31] and – M-learning [1]. Table 5.2 shows the various criteria we used to assess the performance of the three systems. As far as ontology is concerned, unlike the two other systems, our approach uses a global ontology space giving the system a reasoning power by referring to a unique domain space that is homogeneously used by the inference engine. As for the ontology reasoning criterion, our system makes use of the various context groups such as learner context, device context, environment context, and activity context. In particular, learner context is used to represent provisioning personalization. Device context is the main source for representing the software and hardware capabilities of used device. Environment context deals with temporal and spatial contextual information. Unlike the other two systems, our system is characterized by the use of activity context as the basis for personalizing the learning path by tracking and analyzing previous learner's activities. Thus, our system uses activity context to help improve learning content adaptation. For the system-centric adaptations, unlike the other two systems, our system adopts a fuzzy logic approach in conjunction with SWRL rules to translate context that is perceived with uncertainty to meaningful symbolic values. In our case, this is used to predict the maximum data file size and the supported media type(s) based on current bandwidth, current memory size, and screen resolution of the used device. The inference about search language generates suitable information for retrieving learning resources suitable to the learner's linguistic preference. In addition, our system avoids those learning resources that cannot be supported by the software applications on the used device. However, the similarity with the other two systems is that all systems provide some sort of learner-centric adaptations to generate personalized learning paths.

	Proposed mobile learning system	Context-aware E-learning [31]	M-Learning[1]

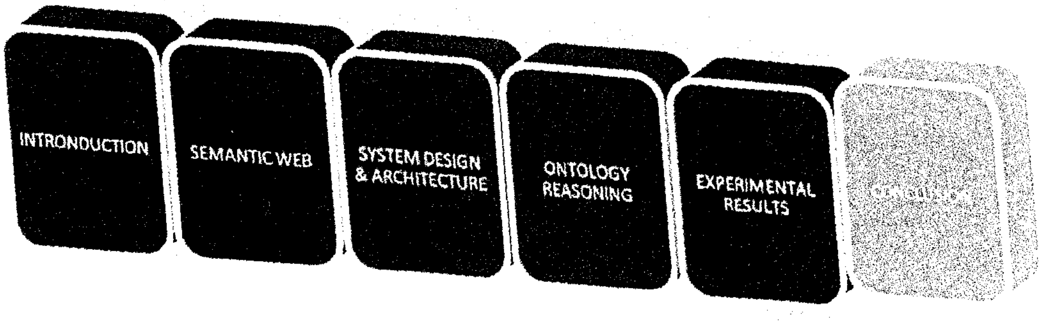
Context Modeling	Global ontology Space: Learner; Activity; Device; Environment; Domain	Learner; Learner Content; Domain	Learner; Device;
			Connectivity;

Ontology Reasoning	System-Centric Adaptations and Learner-Centric Adaptations	Generate Learning Path :Prerequisite	Learning Path: Is-a; Part-Of; Necessary Part-Of Time issue
Inference Technique	SWRL and Fuzzy Logic	Rules	

Table 5.2 Our System v.s. Other Two E-Learning Systems

CHAPTER 6

CONCLUCTION



In this thesis, a personalized mobile learning system on the semantic web has been developed. In particular, an attempt has been made to solve some of the challenges related to context modeling and management; conceptual knowledge modeling for personalized learning; and context-aware service discovery and adaptation. Atomic context is acquired and classified into learner context, device context, environment context, and activity context. These types of context are either sensed or profiled. Sensed atomic context is dynamic in nature, while profiled atomic context is mostly static. The system used fuzzy logic to predict current network bandwidth, allowable file-size, and the appropriate media-type, in order to retrench the service's and resource's expenses. A global ontology space is used to aggregate the above-mentioned context groups which are defined at the semantic level. The role of the global ontology is to integrate a subject-domain ontology along with the learner ontology, activity ontology, device ontology, and environment ontology. Knowledge embedded in the global ontology space is used as the main source to enable a unified reasoning mechanism that operates on facts instantiated by the perceived heterogeneous context elements. In particular, the reasoning engine translates context changes into new adaptation constraints in the operating environment, thus enabling personalized learning. Both system-centric adaptations and learner-centric adaptations have been considered in this study for better personalization of the learning sequence. System-centric adaptations aim at filtering out those learning resources that can run efficiently on the used mobile device, taking into account the attributes characterizing the surrounding environment. The learner-centric adaptations however, aim at building a personalized learning path based on learner's current activity and profile. A number of learning scenarios have been used to demonstrate the main functions of the proposed system. The experimental results have shown that the system successfully adapts the media-type, file size, and other system-centric features, based on the used technology and surrounding environment. The results have also shown successful use of the various learner-centric adaptations to accommodate learners' background and needs. In particular, the system has been tested on two subject-domain ontologies using three different mobile devices. These experiments were conducted under various system environments.

This research work can be extended in many ways. One possible extension is the use of Mashup technology to make it possible to use multiple search agents in order to retrieve learning resources from multiple sources, thus enhancing learning-content provision. Other possible extension to our work is to improve the systems' Quality-Of-Service (QoS). For example, it is important to provide secured services, especially, when moving from one wireless network to another. Trusted web-services are crucial for mobile learning applications such as those related to telemedicine or corporate learning. One further research direction is to use learning paths adopted by various users to build expert knowledge for navigating the subject domain.

REFERENCES

[1] Bssa Basaeed, Jawad Berri, Rachid Benlamri and Jamal Zemerly, "M-Learning Activity-based Context Management", *Digital Information Management, ICDIM '07. 2nd International Conference on Digital Information Management*, October Lyon, France, 2007, pp.701-706

[2] Aroyo, L., and Dicheva, D., "The new Challenges for E-Learning", *The educational Semantic Web, Educational Technology and Society Journal*, 7(4), 2004, pp.59-69

[3] Chittaro, L, "Visualizing Information on Mobile Devices", *IEEE Computer*, 2006, pp.40-45

[4] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications", *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, USA, December 8-9 1994, pp.85-90

[5] Richard, G. and Hatala, M. "Linking Learning Object Repositories", *International Journal of Learning and Technology*, 1(4), 2005, pp.398-409

[6] P. Bellavista, A. Corradi and R. Montanari, "A Mobile Computing Middleware for Location and Context aware Internet Data Services", *ACM Transactions on Internet Technology*, 6(4), 2006, pp.356-380.

[7] V. Derballa and K. Pousttchi, "Extending knowledge management to mobile workplaces", *Proc. of 6th Int. Conf. on Electronic Commerce*, Delft, Netherlands, 2004, pp.583-590.

[8] Stojanovic, L, Staab, S., and Studer, R., " E-Learning based on the Semantic Web", *Proc. of the World Conf. On the WWW and Internet (WebNet2001)*, Orlando, Florida, USA, 2001, pp.1174-1183

[9] Joelle Coutaz, James L. Crowley, "Context is Key", *Communications of the ACM*, March 2005, pp.49-53

[10] Tim Berners-Lee, Hendler, Lassila, "The Semantic Web", 17 May, 2001, Available at: http://www.ryerson.ca/~dgrimsha/courses/cps720_02/resources/Scientific%20American%20The%20Semantic%20Web.htm

[11] J. Davies, M. Lytras and A. Sheth, "Semantic Web Based Knowledge Management", *IEEE Internet Computing*, No. 10, 2007, pp.14-16.

[12] Cardoso, J., Sheth, Amit, "Semantic Web Services", Springer. 2006. ISBN 0-38730239-5

[13] Frank Manola, Eric Miller, Brian McBride, Hewlett-Packard Laboratories, "RDF Primer", *W3C Recommendation*, 10 February, 2004, Available at: http://www.w3.org/TR/rdf-primer/

[14] Michael K. Smith, Chris Welty, Deborah L. McGuinness, "OWL Web Ontology Language Guide", *W3C Recommendation*, 10 February, 2004, Available at: http://www.w3.org/TR/owl-guide/#owl_Class

[15] Anjum Shehzad, Hung Q. Ngo, "Formal Modeling in Context Aware Systems", KI-*Workshop Modeling and Retrieval of Context (MRC2004),* University of Ulm, Ulm, Germany, September 23-27, 2004, pp.13-24

[16] Bijan Parsia, Evren Sirin, Bernardo Cuenca Grau, "Cautiously Approaching SWRL", *Technical Report*, University of Maryland, MIND Lab, 23 February, 2005

[17] Baadar, F., Horrocks, I., Sattler, U.; "Description Logics", *Handbook on Ontologies*, 2004, pp.3-28

[18] Franz Baader, Ian Horrocks, Ulrike Sattler, "Description Logic as Ontology Languages for the Semantic Web", *Number 2605 in Lecture Notes in Artificial Intelligence*, Springer, 2005, pages 228-248

[19] Lan Horrocks, Peter F, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *W3C Member Submission*, 21 May 2004, Available at: http://www.w3.org/Submission/SWRL/

[20] McCalla, G., "The Ecological Approach to the Design of E-Learning, Environments: Purpose-based Capture and Use of Information about Learners", *Journal of Interactive Media in Education*, (7), 2004, pp.1-23

[21] Aroyo, L., and Dicheva, D., "Courseware Authoring Tasks Ontology", *Proc. Int. Conf. on Computers in Education*, Los Alamitos, CA, UAS, 2002, pp.1319-1321

[22] Y. Atif, R. Benlamri and J. Berri, "Dynamic Learning Modeler", *IEEE Journal of Educational Technology and Society*, 6(4), 2003, pp.60-72

[23] Hatala, M., Wakkary, R. and Kalanatari, L, "Ontologies and Rules in Support of Real-time Ubiquitous Application", *Journal of Web semantic*, Special Issue on "Rules and Ontologies for Semantic Web", 3(1), 2005, pp.5-22

[24] Eleni, Christopoulou, A. Kameras, "GAS ontology: an ontology for collaboration among Ubiquitous Computing Devices", *International Journal of Human-Computer studies*, Vol.62, 2005, pp.664-685

[25] Mizoguchi, R, and Bourdeau, J., "Using Ontological Engieering to Overcome Common AI-ED Problems", *Int. Journal of Artifical Intelligence in Education*, 11(2), 2000, pp.107-121.

[26] Guanling Chen, David Kotz, "A Survey of Context-aware Mobile Computing Research", *Technical Report TR2000-318*, Dartmouth College, Computer Science, Hanover, NH, November 2000.

[27] Winograd T., "Architectures for Context", *Human-Computer Interaction (HCI) Journal*, 16(1), 2001.

[28] Nicola Henze, Peter Dolog, Wolfgang Nejdl, "Reasoning and Ontologies for Personalized E-Learning in the Semantic Web", *Educational Technology & Society*, 7 (4), 2004, pp.82-97.

[29] Ki-Chul Lee, Kim, Jee-Hyong Lee, "Implementation of Ontology Based Context-Awareness Framework for Ubiquitous Environment", *International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, 2007, pp.278-282

[30] Stephen J.H.Yang, "Context Aware Ubiquitous Learning Environment for Peer-to-Peer Collaborative Learning", *Educational Technology & Society*, 9(1), 2006, pp.188-201.

[31] Zhiwen Yu, Yuichi Nakamura, "Ontology-Based Semantic Recommendation for Context-Aware E-Learning", J. Indulska et al. (Eds): Springer-Verlag Berlin Heidelberg , UIC 2007, in *Lecture Notes in Computer Science*, Vol. 4611, 2007, pp.898-907.

[32] S. Braun and A. Schmidt, "Don't Annoy the Informal Teacher: Context-Aware Mediation of Communication for Workplace Learning", *Proc. Int. Conf on Knowledge Management, Graz*, In Klaus Tochtermann, Hermann Maurer (eds.). Springer Verlag, 2006, pp.429-436.

[33] R. Denaux, V. Dimitrova and L. Aroyo, "Integrating Open User Modeling and Learning Content Management for the Semantic Web", Proc. 10th Int. Conf. on User Modeling, Edinburgh, UK, In *Lecture Notes in Computer Science*, Vol. 3538, 2005, pp.9-18.

[34] Dolg, P., Henze, N., Nejdl, W., & Sintek, M, "The Personal Reader: Personalizing and Enriching Learning Resources Using Semantic Web Technologies", *Lecture Notes in Computer Science*, Vol.3137, 2004, pp. 85-94

[35] Jawad Berri, Rachid Benlamri, and Yachine Atif, "Ontology-based Framework for Context-aware Mobile Learning", *Int. Workshop Context-aware Mobile Computing '06*, Vancouver, Canada, July 3-6, 2006, pp.1307-1310

[36] Panu Korpipaa, Jani Mantyjarvi, "Managing Context Information in Mobile Devices", *IEEE CS and IEEE ComSoc*, 2003, pp.42-51

[37] Jeff Z.Pan, Giorgos Stoilos: f-SWRL, "A Fuzzy Extension of SWRL", *Lecture Notes in Computer Science*, Vol.3697, 2005, pp.829-834

[38] J. Mantyjarvi and Seppanen, "Adapting Applications in Handheld Devices Using Fuzzy Context Information", *Interacting with Computers* J*ournal.*, vol. 15, No. 4, 2003, pp.521-538

[39] Quan Thanh Tho, Siu Cheung Hui, "Automatic Fuzzy Ontology Generation for Semantic Web", *IEEE Transactions on Knowledge and Data Engineering*, pp. 842-856

[40] Rachid Benlamri, Xiaoyun Zhang, Jawad Berri, "Proactive Mobile Learning on the Semantic Web", *Int. Workshop on Ubiquitous Computing WUC'07*, Portugal, June 2007, pp.63-73

[41] R. Benlamri, J. Berri and Y. Atif, "A Framework for Ontology-aware Instructional Design and Planning", *Journal of E-Learning and Knowledge* Society, 2(1), 2006, pp.83-96.

[42] R. Benlamri, Y. Atif, and J. Berri, "An Ontology-based Approach for Context-aware Mobile Learning", *In Advances in Ubiquitous Computing: Future Paradigms and Directions*, IGI Global Ed., 2008, pp.23-44.

[43] P. Dolog and W. Nejdl, "Semantic Web Technologies for the Adaptive Web, I Adaptive Web: Methods and Strategies of Web Personalization", *Lecture Notes in Computer* Science, Vol. 4321, 2007, pp.697-719.

[44] R. Krummenacher, J. Kopecky and T. Strang, "Sharing context information in Semantic Spaces", *Lecture Notes in Computer Science*, Vol. 3762, 2005, pp.229-232

[45] Fred Cooper, "Adapting to Available Network Bandwidth", Intel ® Software Network, Mobilizing Applications, 28 February, 2007, Available at: http://softwarecommunity.intel.com/articles/eng/1074.htm

APPENDIX A

PUBLISHED PAPERS

- Proactive Mobile Learning on the Semantic Web, Proc. Of the 4th International Workshop on Ubiquitous Computing-WUC 2007, Funchal-Madeira, Portugal, 12-16 June, 2007, pp.63-73.

- A Global Ontology Space for Mobile Learning, To appear in Proc. of 8th IEEE International Conference on Advanced Learning Technologies –ICALT'2008, Paper ID 385, to be published by IEEE Xplore.

Proactive Mobile Learning on the Semantic Web

Rachid Benlamri[1], Jawad Berri[2] and Xiaoyun Zhang[1]

[1]Department of Software Engineering, Lakehead University
955 Oliver Rd, Thunder Bay, Ontario, P7B 5E1, Canada
{rbenlamr, zhang6}@lakeheadu.ca
[2] Department of Computer Engineering, Etisalat University College
PO Box 573 Sharjah, United Arab Emirates
berri@euc.ac.ae

Abstract. Flexible and personalized instruction is one of the most important requirements to next generation intelligent educational systems. The intelligence of any e-learning system is thus measured by its ability to sense, aggregate and use, the various contextual elements to characterize the learner, and to react accordingly by providing a set of customized learning services. In this paper we propose a proactive context aware mobile learning system on the Semantic Web. The contribution of this work is a combined model using both a probabilistic learning technique and an ontology-based approach to enable intelligent context processing and management. The system uses a Naïve Bayesian classifier to recognize high level contexts in terms of their constituent atomic context elements. Recognized contexts are then interpreted as triggers of actions yielding a Web service composition. This is achieved by reasoning on the ontological description of atomic context elements participating in the high level context.

1 Introduction

Research work in the field of mobile learning [1][2][3][4][5] has shown that the educational potential of mobile technologies is driven by the continuing expansion of broadband wireless networks and the capacity of the new generation of cellular phones. However, the utilization of these technologies for educational purposes has been sparsely explored and many problems related to: context acquisition and management, conceptual knowledge modeling for personalized instruction, and adaptive information discovery remain unresolved. This paper contributes towards this direction, aiming at using the evolving semantic web and mobile computing to enable context-aware learning which delivers adaptive instructional resources on a learner's schedule. Context-aware learning is a critical support mechanism for educational institutions and organizations to compete in the new economy. Today's global market requires adaptive, fast, just-in time, and relevant learning processes that can be initiated by user profiles and business demands [6].

In this paper we propose an integrated approach to context modeling and reasoning based on Naïve Bayesian classifiers and ontological structures. First, higher-level contexts are recognized using a Naïve Bayesian classifier. Then, ontology-based reasoning with the recognized contexts triggers actions yielding Web service composition that are customized to learner's context, needs, and preferences. The contextual information used in the personalization process encompasses all elements that characterize the learner's interaction, task at hand, the resources on which the Web services are to be performed, and surrounding environment.

The remaining of the paper is organized as follows. Section 2 describes background knowledge and related work. Section 3 describes context representation and modeling schemes. In section 4, we describe the higher-level context recognition process. Section 5 presents the framework for ontology reasoning and Web services composition to generate adaptive learning services. Finally, conclusions are drawn and further research work is suggested.

2 Background and Related Work

A considerable amount of research in knowledge-based and intelligent e-learning systems is now moving towards ontology-based context acquisition and management for personalized learning [7][8][9][10]. The main issues and challenges are however related to the ability of such systems to model and consistently reason with high level contexts at the semantic level. Although, some research attempts were made to solve some of these problems [9][10][11][12], the shortcoming of most of these efforts is their limitations to specific context elements and specific learning scenarios. General-purpose modeling and reasoning with context is a complex problem, and much research work is needed before achieving any real progress in this field. Most developed learning systems restrict the use of ontology relations and rules to describing and adapting content and sequencing of learning material according to some sensed context. However, little contextual semantics has been embedded in the ontology itself.

Other approaches to context modeling have also been considered. McCalla [13] has introduced an approach to learning design where learners' models are attached to Learning Objects (LOs) they interact with, and useful learning patterns are then derived by mining those models. The problem with this approach is its limitation to context that can be inferred from the learner's profile only, ignoring other type of context. Stojanovic et al. [6] however, have extended ontology usage to describe content, context, and sequence of learning material. Content-ontology was used for checking consistency as well as searching and navigating repositories of LOs. Context-ontology was used to present learning material in various learning contexts. However, learning style ontology was used to describe the way knowledge can be dynamically connected to adapt to learners' cognitive needs and preferences. Sets of relations, rules and axioms have been separately defined for each type of adaptation. The shortcoming of this approach is that efficient modeling of mobile learning scenarios would require the definition of atomic context elements at the semantic level and the use of the various ontologies in an orthogonal way. This is due to the fact that context, content, and learning styles are semantically inter-related aspects of cognitive learning [14]. This paper explores such a new dimension. The emphasis is on context discovery and its semantic modeling and management. Mobile users equipped with wireless devices go through several contextual changes as they move around in physical and social surroundings. These contextual changes could be used to drive ontology navigation and reasoning for better modeling of mobile learning scenarios.

Another challenging aspect addressed in this paper is automation of metadata generation for mobile learning. Metadata provides a common set of tags for describing, indexing, searching, and reusing learning materials on the Web in an interoperable way [14]. However, it is really difficult to create and maintain metadata rich enough to meet the diverse and ever changing needs of potential mobile learners. Mobile learning requires additional metadata to capture context. In this study, an attempt is made to solve this problem by defining contextual information at three hierarchical levels – atomic context – composite context – and higher-level context. Atomic context elements are sensed from the learner's interaction, task at hand, the used mobile-device, and the surrounding environment. These are then grouped into four composite context classes – learner context – activity context – device context and environment context. Composite contexts are further aggregated to build meaningful time-stamped higher-level contexts which are matched against context classes describing typical learning scenarios. Context classes are simply built from previously sensed similar higher-level contexts that have exhibited high degree of confidence. Matching higher-level contexts against these context classes is performed using a Naïve Bayesian classifier. The Naïve Bayesian classifier technique is used to cope with the uncertainty embedded in most sensed atomic contextual elements. Recognized contexts are then interpreted as triggers of actions that are translated into Web service compositions. This is achieved through ontological descriptions and reasoning with higher-level context.

Fig. 1 describes the overall system architecture which consists of four main components – context acquisition and aggregation – context recognizer – ontology reasoning engine – and Web-service composer. The context acquisition and aggregation component controls the user's interaction with the system and senses atomic context information from different sources. These are then aggregated into domain related contexts. Mobile learners go through continuous contextual changes as they move in their environments. It is the context acquisition and aggregator's job to communicate and update such changes yielding new contexts. The context recognizer identifies the aggregated contexts by matching them against well defined context classes stored in a context repository. The recognition process is performed using a Naïve Bayesian classifier. The context recognizer also allows for newly formed context classes to be added to the context repository.

The third component of the system is an ontology reasoning engine which uses the recognized higher-level contexts to customize learning services. Two ontologies are used to perform such a task – device/environment ontology – and domain ontology. The former is used to generate metadata that is used to discover Web-services that can run in the learner's device/network environment. However, the later is used to customize the learning content and the learning sequence according to the learner's current activity, background and preferences. This requires an ontological description and interpretation of higher-level contexts in terms of their constituent atomic context elements. Finally, the Web-service composer uses the generated device/environment metadata and the inferred learning concepts' sequence to compose Web-services accordingly.

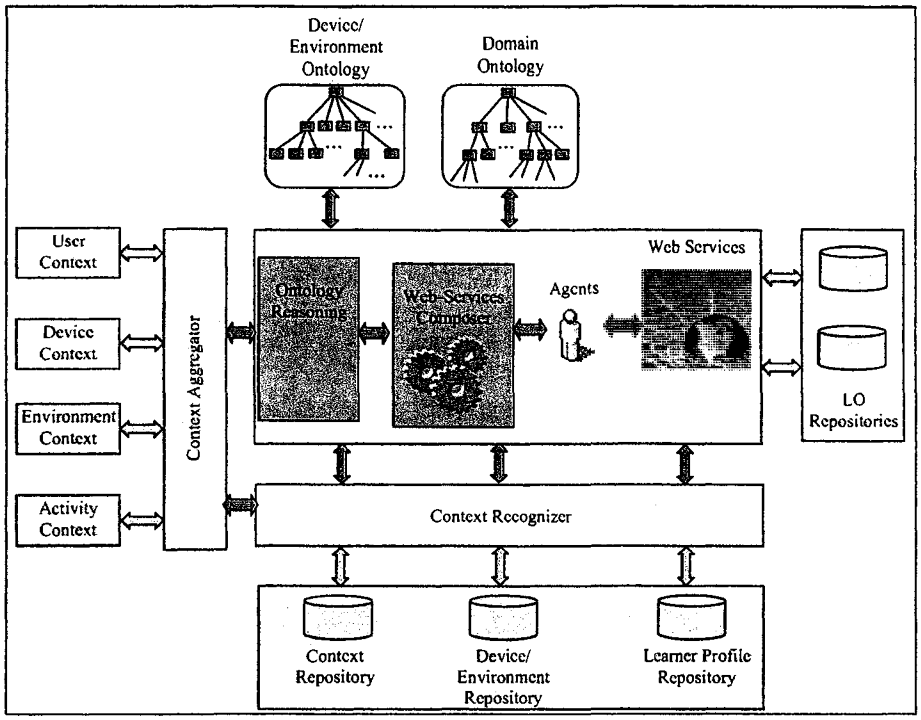


Fig. 1. System Architecture.

3 Context Acquisition and Aggregation

Contextual information used in this study is defined at three hierarchical levels – atomic context – composite context – and higher-level context. At the lower level, atomic contextual elements consist of the basic information describing the learner's profile, the current learner's activity, the used mobile device, and the surrounding environment. These can be either direct or indirect atomic contextual elements. Direct atomic contextual elements are those that can be directly sensed from the user interaction with the system and may originate from different sources such as the used device (i.e. device type, communication protocol), the task at hand (i.e. current learner's activity), and the surrounding environment (i.e. location, time, wireless network, network security). Indirect atomic contextual elements are however those elements that can be indirectly inferred from the direct atomic context elements. Inference of indirect atomic context elements is performed by the context aggregator relying on the device/environment repository and the learner profile repository. For instance, information such as device's operating system, device memory, and screen resolution of a specific mobile device which is previously stored in a device repository can be inferred using the atomic context element *device-type*. Similarly, other information related to the learner's pre-requisite knowledge, previously accessed services, and learner's preferences can be inferred from the learner profile repository. The use of indirect contextual elements aims at reducing the amount of contextual information that has to be sensed from the learner's interaction, device, and surrounding environment, which significantly speedup the context recognition process.

An atomic context element c_i is defined by:

$$c_i = (c_{iv}, c_{ip}) \tag{1}$$

where c_{iv} is the context value, and c_{ip} is the probability of context c_i of value c_{iv} being part of a higher-level context. The context value c_{iv}, as shown below, can be either a specific value (i.e. device type, learner identifier), a binary value (i.e. whether the used device is browser-enabled or not, secured/non-secured wireless network), or a value within a predefined range (i.e. network bandwidth, screen resolution).

$$c_v = \begin{cases} specific\_value \\ binary\_value \\ value \in [v1..v2] \end{cases} \tag{2}$$

Composite contextual elements are aggregates of atomic context elements describing a specific context type. There are four context types – c_L learner context – c_D device context – c_E environment context – and c_A activity context. Each of which is defined by:

$$c_{composite} = \left\{ \sum_{i=1}^{p} c_{i-direct} \cup \sum_{j=1}^{q} c_{j-indirect} \right\} \tag{3}$$

Finally, higher-level contexts consist of four-tuples $C_t = (c_L, c_D, c_E, c_A)_t$ which are built out of configurations of composite context elements sensed at time *t* and which characterize typical learning scenarios in a specific domain. Classes of higher-level contexts are defined at the ontological level in that they can be interpreted directly as triggers of learning actions implemented as Web service compositions.

4 Context Recognition

While ontologies have the ability to communicate context information by naming different concepts in machine readable fashion and allowing for the use of everyday words and concepts when interacting with the technology, they are unable to efficiently recognize learners' context. This is because the mapping between the defined concepts and the sensed real world atomic context elements is not so straightforward due to the *uncertainty* embedded in some atomic context elements. The mapping fails because ontologies do not handle *uncertainty*. They rather rely on well defined logic which assumes all information required to make a logical decision is available and produces either true, false or undeterminable statements. Uncertainty on the other hand produces similar statements but with degrees of truth or falseness [15]. To cope with uncertainty, higher-level contexts are recognized using a Naïve Bayesian Classifier. Bayesian Classification is a probabilistic learning technique where prior knowledge can be combined with observed data. The aim is to recognize a currently observed context state against a set of learned context classes. The input to the classifier is thus a set of sensed/observed atomic context elements which describe the user's context at a given instant of time, while the output is a learned context class. The classification process is thus performed with no user intervention or understanding required. The Bayesian classification also makes the implicit assumption that the data being handled is noisy and can tolerate any missing pieces of information. One difference between the Bayesian classification and the ontology approach is that once the ontology is defined then it can be available immediately whereas in the Bayesian classification approach each context has to be experienced at least once before being recognized again [15].

Let X be a current context whose class label is unknown, and let H be a hypothesis that X belongs to context class C, the classification problem consists of determining $P(H/X)$ that is the probability that the hypothesis holds given the observed context X. This is defined by:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \tag{4}$$

where :

- $P(H)$ is the prior probability of hypothesis H (i.e. the initial probability before we sense the current context and reflects the background knowledge).
- $P(X)$ is the probability associated to the current context.
- $P(X|H)$ is the probability of observing the context X, given that the hypothesis holds.

The above Bayesian model assumes that the observed context elements are related and depend on each other, and therefore, requires initial knowledge of many probabilities, as well as, significant computational cost. However, since most sensed atomic context elements are independent, the above model can be further simplified by applying the Naïve Bayesian classifier which is defined by:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{5}$$

Where: C_i is a context class, and the set of x_ks are the atomic context elements forming the higher-level context X as defined in section 3.1.

The Naïve Bayesian classifier greatly reduces the complexity of the model, as well as its computational requirements. The context recognition problem is solved by assigning the current context X to the class C_k that satisfies the following condition:

$$X \in C_k \left| P(X|C_k) = \underset{i=1..m}{Max}\{P(X|C_i)\cdot P(C_i)\}\right. \tag{6}$$

where m is the number of recognized context classes.

Fig. 2 describes the context acquisition and recognition cycle. First, direct-atomic context elements are sensed, these are then used to infer related indirect-context elements. Next, the Naïve Bayesian classifier is applied to recognize the associated higher-level context-class, and finally, changes to the learner's context are sensed and a new context recognition cycle is performed. It should be noted here that the context-change detection process significantly speedup the recognition time of successive high level contexts. This is because we just infer the indirect-atomic contexts of those context elements that have undergone some changes. The subset of newly observed context elements designated by $C_{changes}$ is defined by:

$$C_{changes} = (c_L, c_D, c_E, c_A)_{t_{i+1}} \setminus (c_L, c_D, c_E, c_A)_{t_i} \qquad (7)$$

where "\" means set subtraction.

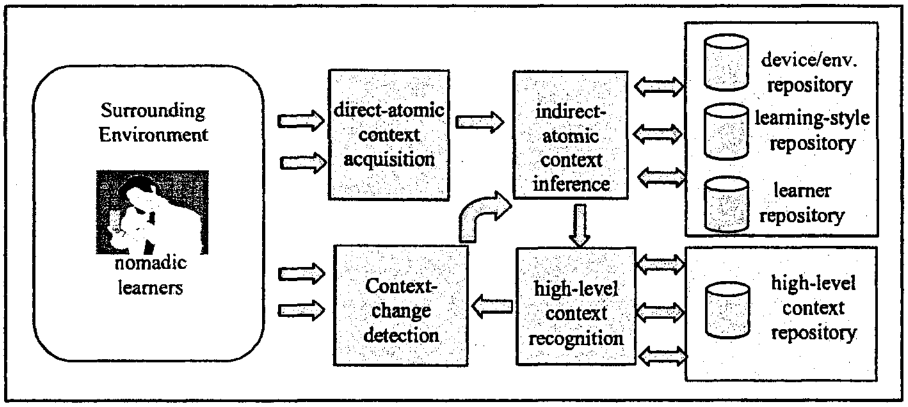

Surrounding Environment
nomadic learners
direct-atomic context acquisition
indirect-atomic context inference
device/env. repository
learning-style repository
learner repository
Context-change detection
high-level context recognition
high-level context repository

Fig. 2. Context Acquisition and Recognition Cycle.

5 Ontology Reasoning and Web Service Composition

Recognized higher-level contexts are fed to the ontology reasoning engine in order to customize learning services based on the learner's context, preferences and background. Reasoning with recognized higher-level contexts is performed using the two ontologies – device/environment ontology – and domain ontology. A set of ontological rules are applied to the device/environment ontology to infer the computing resources and the operational environment features compatible with the used mobile device and its surrounding environment. We call this process, *context-driven resources adaptation.* The output of this reasoning process is a set of metadata that will help discovering the Web services that can run into such an operational environment. The inference rules that are built around the domain ontology however are used to provide the learner with a learning sequence and content tailored to his/her current activity, previous background and preferences.

The two ontologies are coded in the Web Language Ontology – OWL; and the inference engine is implemented in Rule Markup Language – RuleML. Metadata derived from the ontology reasoning process is compliant with the IEEE-LTSC Learning Object Metadata (LOM) specification which is coded in XML. In particular, the XML description of both the inferred learning concepts and the device-related operational environment are used for Web services discovery. However, the inferred learning sequence which we call in this paper *domain-context* (i.e. the order of learning concepts inferred using the properties and relations between the domain-ontology's classes [16]) is used for Web service composition. This is described in OWL-S. A domain context represents a control structure that makes it possible to adapt the domain knowledge to a particular higher-level context. This adaptation is facilitated by the ontology O_M for a given domain M. O_M is defined by $O_M = (Co_M, R_M)$, where $Co_M = \{co_1,...,co_n\}$ is a set of concepts and $R_M = \{r_1,...,r_q\}$ is an ordered set of rules defined as follows: $p(co_1...co_j) \rightarrow_{r_s} q(co_k...co_l)$, where p and q are predicates reflecting respectively the factual information and the resulting one based on the inferential rule r_s.

The semantic of the ontological links is obtained by the rules in R_M. These rules are prioritized to reflect their importance or abstraction levels in a given knowledge taxonomy. For example, if the sensed higher-level context reflects a time-constrained learning scenario, one would like to focus only on say "the necessary-part-of" rules of the ontology to get a quick abstraction on the general structure of the requested knowledge. In a less time-stringent learning scenario however, this abstraction could further include the "part-of", and/or "case-study" rules, etc. These knowledge-supporting rules generate additional concepts of the ontology in multi-level clusters which are used to infer a progressive knowledge based on the learners' context denoted by C_L and the activity context denoted by C_A as described in section 3.

A software agent as shown in Fig. 1 is spawned at the server side to supervise a learning session for each learner. The agent typically represents the learner on the Semantic Web. The agent successively invokes the inference engine to get the current learner's focus, then discovers, composes, and invokes the chosen Web services accordingly.

To illustrate the main functions provided by our framework, we provide the following example ontologies describing a C++ programming course as a domain ontology, and a device/environment ontology. These are shown in Fig. 3 and Fig. 4 respectively.

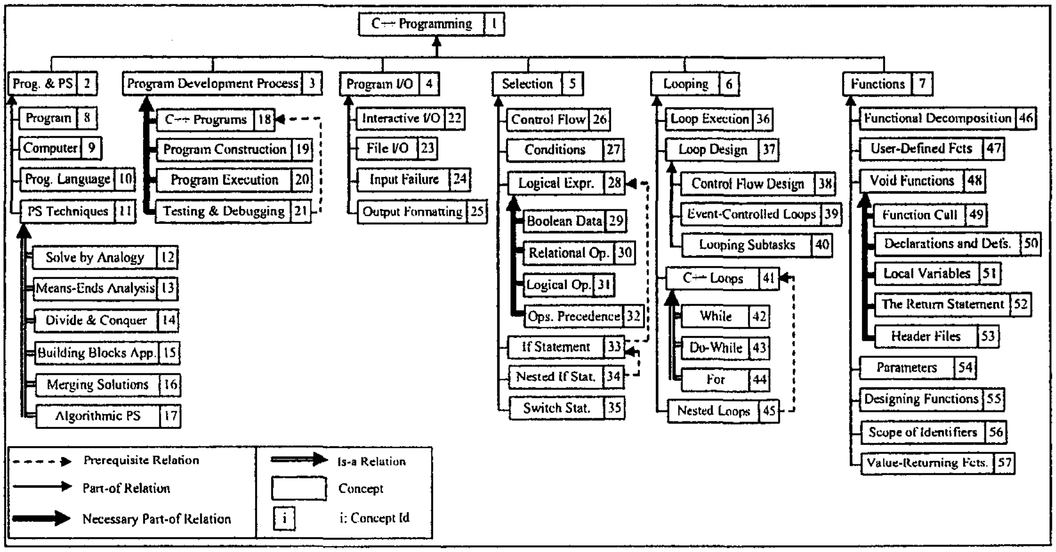


Fig. 3. Ontology for C++ Programming Course.

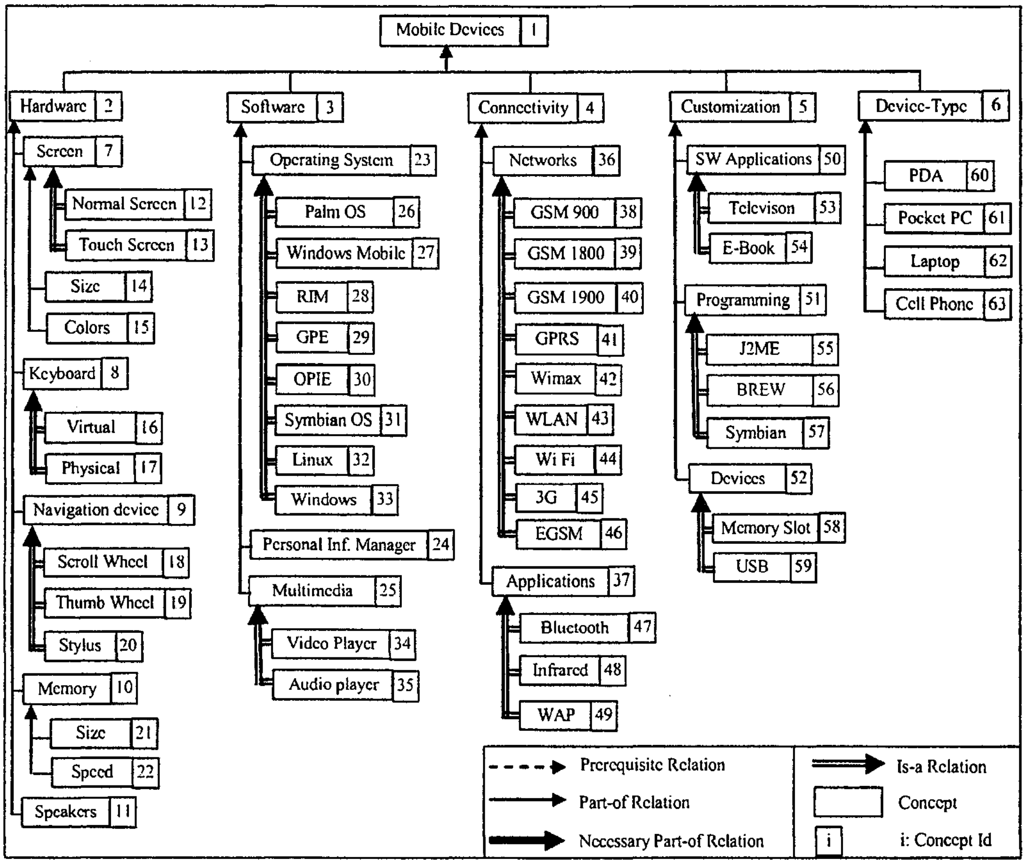


Fig. 4. Device/Environment Ontology.

A fragment of the ontology shown in Fig. 3, describing concept 3 "*Program Development Process*", is described in OWL in Fig. 5. The OWL definition of the semantics of the different relationships used in the C++ programming ontology is also given in Fig. 5.

Details about the rules used by the ontology reasoning engine to customize the learning sequence can be found in our previous work [17].

```
<owl:ObjectProperty rdf:ID="NecessaryPartOf">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <owl:inverseOf rdf:resource="#hasNecessaryPart"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isPartOf">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <owl:inverseOf rdf:resource="#hasPart"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="is-a">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <owl:inverseOf rdf:resource="#has"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isPrerequisiteOf">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
     <owl:inverseOf rdf:resource="#hasPrerequisite"/>

<owl:Class rdf:ID="Program Development Precess_3">
  <rdfs:subClassOf rdf:resource="#C++ Programming_1"/>
  <owl:disjointWith rdf:resource="#Prog and PS_2"/>
  <owl:disjointWith rdf:resource="#Program I/O_4"/>
  <owl:disjointWith rdf:resource="#Selection_5"/>
  <owl:disjointWith rdf:resource="#Looping_6"/>
  <owl:disjointWith rdf:resource="#Functions_7"/>
</owl:Class>

<owl:Class rdf:ID="C++ Programs_18"/>
  <rdfs:subClassOf rdf:resource="#Program Development Process_3"/>
  <rdfs:subClassOf>
    <owl:Restriction>
          <owl:onProperty rdf:resource="#isNecessaryPartOf"/>
          <owl:allValuesFrom rdf:resource="#program Development Process_3"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
          <owl:onProperty rdf:resource="#isPrerequisiteOf"/>
          <owl:allValuesFrom rdf:resource="#Testing and Debugging_21"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Program Construction_19"/>
  <owl:disjointWith rdf:resource="#Program Execution_20"/>
  <owl:disjointWith rdf:resource="#Testing and Debugging_21"/>
   </owl:Class>
```

Fig. 5. Fragments of OWL description of the C++ Programming ontology.

6 Conclusions

In this paper, we proposed a proactive mobile-learning system on the Semantic Web. We argued that a probabilistic learning model is more suitable that an ontology-based approach for context recognition. This is mainly due to uncertainty embedded in some atomic contextual information. Higher-level recognized contexts are however described at the semantic level using ontology rules and axioms. The ontology reasoning process allows the system to react to any observed contextual changes by interpreting the newly sensed contexts as triggers of actions yielding a Web service composition. We are currently implementing a prototype of our framework as part of our personalized-learning provision project.

Acknowledgements

This work is funded by Lakehead University and NSERC under a startup research grant.

References

1. L. Low and M. O'Connell, "Learner-centric design of digital mobile learning", Proc. of the On-line Learning and Teaching Conference, Brisbane, Australia, September, (2006) 71-82.
2. M. Milrad, "Mobile Learning: Challenges, Perspectives and Reality", Book chapter in Kristóf Nyíri (ed.), Mobile Learning: Essays on Philosophy, Psychology and Education, Vienna, (2003) 25-38.
3. B. Alexander, "Going Nomadic: Mobile Learning in Higher Education", EDUCAUSE Review, 39(5), (2004) 28–35.
4. L. Chittaro, "Visualizing Information on Mobile Devices", IEEE Computer, March (2006) 40-45.
5. G. Vavoula and M. Sharples, KLeOS: A personal Mobile Knowledge and Learning Organisation System", Proc. 2nd IEEE Int. Workshop on Wireless and Mobile Technologies in Education, Sweden, (2002) 152-157.
6. L. Stojanovic, S. Staab and R. Studer, Elearning based on the semantic web, Proc. of the World Conf. on the WWW and Internet (WebNet2001), Orlando, Florida, USA, (2001) 1174-1183.
7. R. Mizoguchi and J. Bourdeau, Using Ontological Engineering to Overcome Common AI-ED Problems", Int. Journal of Artificial Intelligence in Education, 11 (2), (2000) 107-121.
8. L. Aroyo and D. Dicheva, Courseware Authoring Tasks Ontology", Proc. Int. Conf. on Computers in Education, Los Alamitos, CA, (2002) 1319-1321.
9. J. Kay and A. Lum, Ontology based User Modeling for the Semantic Web, Proc. of the Int. Workshop on Personalisation on the Semantic Web, Edinburgh, UK, (2005) 14-23.
10. J. Wang et al., "Personalized Knowledge Service Framework for Mobile Learning", Proceedings of the Second Int. Conf. on Semantics, Knowledge, and Grid (SKG'06), November, (2006) 102-103.
11. I. Tirrellil et al., "Fractal Adaptive Web Service for Mobile Learning", Int. Journal, of Emerging Technologies in Learning, 1(2), (2006) 25-35.
12. P. Veith, "Conception and Development of Reusable and Modular Mobile Content", Proc. of Int. Conf. on Mobile Learning mLearn'2005, Cape Town, 25-28 October, (2005), Retrieved from <http://www.mlearn.org.za/CD/ > Dec. 2006.
13. G. McCalla, The Ecological Approach to the Design of E-learning, Environments: Purpose-based Capture and Use of Information about Learners, Journal of Interactive Media in Education, 7, (2004) 1-23.
14. L. Aroyo and D. Dicheva, The new challenges for e-Learning: The educational Sementic Web, Educational Technology and Society, 7(4), (2004) 59-69.
15. J. Flanagan, Unsupervised clustering of symbol strings, in Intl' Joint Conference on Neural Networks,IJCNN'03, Portland Oregon, (2003) 3250–3255.
16. R. Benlamri, J. Berri and Y. Atif, A Framework for Ontology-aware Instructional Design and Planning, Journal of E-Learning and Knowledge Society, 2(1), (2006) 83-96.
17 J. Berri, R. Benlamri and Y. Atif, Ontology-Based Framework for Context-aware Mobile Learning, Proc. Int. Conf. on Wireless Communication and Mobile Computing, Vancouver, Canada, (2006) 1307-1310.

A Global Ontology Space for Mobile Learning

Rachid Benlamri and Xiaoyun Zhang
Department of Software Engineering – Lakehead University
955 Oliver Rd, Thunder Bay, Ontario, P7B 5E1, Canada
{rbenlamr, xzhang6}@lakeheadu.ca

Abstract

In this paper we present a knowledge-driven model for mobile learning based on the semantic web. The knowledge model uses a global ontology space and a unified reasoning mechanism to integrate and aggregate knowledge describing both system-centric and user-centric context information. The reasoning engine perceives, understands, and translates context changes into new adaptation constraints in the operating environment to achieve personalized learning. In particular, the system strives to adapt the learning sequence and the learning content based on the learner's activity, profile, used technology, and surrounding environment. An initial system prototype is described and the obtained experimental results are very promising.

1. Introduction

The field of mobile services is becoming a very active area of research and development [1-2]. However, very little has been accomplished in the area of mobile learning. Several obstacles still hinder personalization of mobile learning services, such as: (i) current mobile web services act as passive components rather than active components that can be embedded with context awareness mechanisms, (ii) existing approaches for service composition typically facilitate choreography only, while neglecting contextual information on users and surrounding environment, and (iii) lack of context modeling techniques and reasoning strategies for integrating the various contextual features for better personalization. In this paper, an attempt is made to solve some of the above mentioned problems, aiming to build a mobile learning system with semantic-rich awareness information.

Semantic Web has the potential to revolutionize the way learning services available on the web are discovered, adapted, and delivered according to context [3-6]. In this paper, we demonstrate such capabilities by proposing a knowledge driven model based on a unified reasoning mechanism and a global ontology space that encompasses all context aspects to achieve personalized mobile learning. In particular, whenever context change occurs, the Run-Time Environment (RTE) identifies the new contextual features and translates them into new adaptation constraints in the operational environment to achieve both user-centric and system-centric adaptations. A prototype system using the above mentioned configuration is being developed and initial results are very promising. The system combines Fuzzy Logic and Semantic Web Rule Language (SWRL) to infer context that is quantized with uncertainty, and that can be inferred from ontology respectively. It also uses *Mashup* technology for service discovery and invocation from different distributed repositories.

The remainder of this paper is organized as follows. Section 2 describes the overall system design and architecture. Section 3 describes the approach used for context sensing and representation. Section 4 describes ontology context modeling and reasoning to achieve personalized learning. We also show some experimental results. Finally, conclusions are drawn and further research is suggested.

2. System Design and Architecture

The core of the proposed system is based on a RTE designed to maintain consistent behavior across variations in the operating environment. The aim is to provide learning services adapted to the learner's global context. Therefore, the main function of the RTE is to coordinate and facilitate integration and fusion of the four main context components as they emerge through the learner's interaction with the system. To achieve such complex task, we structured the RTE into three hierarchical levels. As shown in Figure 1, at the lower level of the hierarchy is the context sensing layer which is provided by a collection of hardware and software sensors that continuously probe the wireless network features, temporal-spatial data, device features, user's background, and preferences. The context sensing layer generates quantized and non-quantized raw data whose values are numeric, Boolean, and literals, and most of which are time-stamped. To transform this context data into meaningful context, the raw-data is translated into symbolic information. The mapping is achieved by the context perception layer through computation, inference and learning techniques. The context perception layer is independent from the context sensing technology in the sense that it provides an abstract context representation through the use of ontologies.

At the higher level of the RTE hierarchy is context identification and adaptation layer where learning services are discovered and learning content is adapted based on the interpreted context. The integrated ontology space describing knowledge about all context components is incremented with domain ontology knowledge, and used as a unified knowledge base for system reasoning. The result of the reasoning process is a set of extracted metadata used for service discovery and adaptation based on system-centric context (device and environment context) and user-centric context (learner and activity context). In particular, the extracted metadata is used to personalize both the learning path and learning content in order to match the learner's background, prerequisite requirements, previous tasks, learner's mobility, available network bandwidth, privacy and connectivity issues. Each of these adaptations is controlled by a *Context-Adaptation Logic* in the form of ontology reasoning steps.

3. Context Sensing and Representation

Context is any information that is relevant to the interactions between a user and an environment [5]. This information is about the circumstances, objects, and conditions by which the user is surrounded. Contextual information can be classified into atomic context and composite context. Atomic context elements are associated to raw data that is either sensed or profiled. Sensed atomic context is mainly dynamic in nature such as user location or network bandwidth. Profiled atomic context however is mainly static such as screen resolution of a specific handheld device, user's date of birth, gender, or preferred language(s). Composite context on the other hand is derived from atomic context elements through computation, inference, or learning techniques.

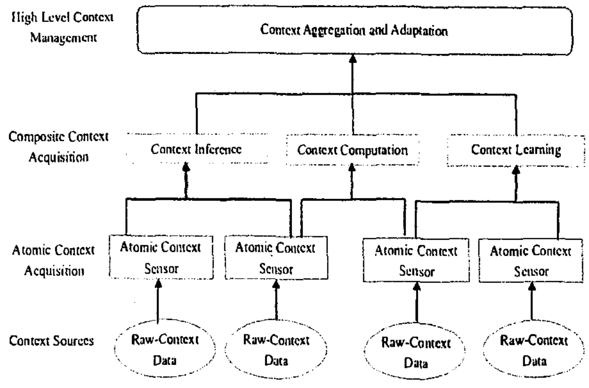


Figure 1. Run-time environment hierarchy

In this study we divide context into four context groups – User context – Activity context – Device context – and Environment context. User context is the main source for provisioning personalization. It also extends activity context by providing information such as user's background, preferred language(s), and user's schedule. Activity context however deals with accessed services, consumed learning resources, adopted learning sequence, and domain-knowledge management and adaptations. It uses learning domain ontology as the main backbone for service adaptation and content management. Device context on the other hand is the main source for determining the software and hardware capabilities of used devices and hence is used for setting the right execution profile for the accessed services. Information such as device operating system, screen resolution, available memory, and supported device applications are crucial to target metadata that allow discovery of services that can run on such devices. Finally, environment context deals with information such as temporal and spatial contextual information, network bandwidth, and other service quality features including security. Environment context extends device context by adjusting the execution profile of accessed services. For example, while choosing the media type of the resources to be retrieved, we may not solely depend on the capabilities of the used device, but we should also take into consideration current network bandwidth.

Ideally, all context changes need to be fed to the system as they occur. However, the process of continuously sensing and updating the dynamic atomic context elements is time and resource consuming, especially in a mobile computing environment where system resources are very expensive. To solve this problem, we adopt an approach where precise values about some of these context elements, such as network bandwidth for instance, are sensed at some specific points in time, and approximate values are predicted with reasonable certainty outside these points. We use fuzzy logic to predict the value of such dynamic contextual elements.

Figure 1 shows the main components of the context sensing and perception layers. At the low level, software and hardware sensors are used to sense and collect atomic context raw-data from different sources. Some of these atomic contexts are sensed and others are retrieved from device and user profiles. Other atomic context such user identification and authentication information are input by the learner. The sensed raw data is then translated into symbolic meaningful context information through inference, computation or learning techniques.

4. Context Modeling and Reasoning

At the semantic level we define contextual information using a global ontology space that integrates the four context ontologies and the subject domain ontology. Context aggregation is enabled using a shared ontology space and a unified reasoning mechanism across these ontologies. In particular, whenever context change occurs, the run-time environment identifies the new contextual features and translates them into new adaptation constraints in the operational environment. Figure 3 shows the global ontology space which encompasses the four context ontologies and the domain ontology. Different types of core ontology classes describing basic ontology concepts (i.e. Device, Learning Resource), role concepts (i.e. Learner), and role holders (i.e. LearningActivity), are used to interrelate concepts among the combined ontologies. As shown in Figure 2, the five ontologies are integrated and blended along the many properties that link various classes used by these ontologies. Figure 2 shows only the relationships among these ontologies. Below, we describe each of these ontologies as well as the relationship between them in more details.

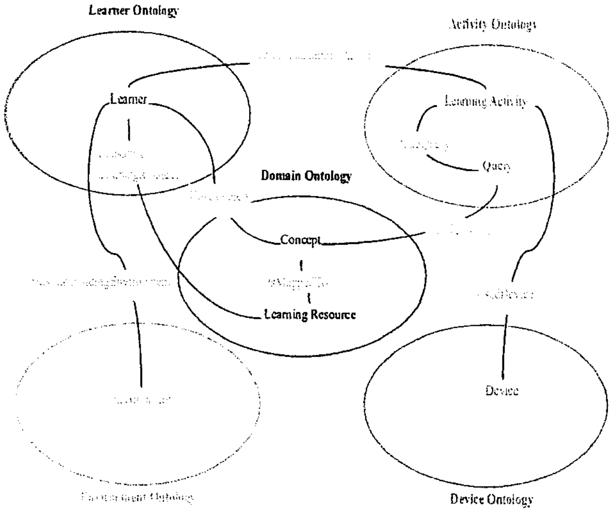


Figure 2. Global ontology space

The domain ontology is a conceptualization of a knowledge organization of a specific subject domain. It is expressed in terms of a hierarchy of subject topics, each of which is described by a set of concepts and their relationships. The power of domain ontology is thus measured by its ability to model the semantical role of its concepts in terms of their importance to the described subject domain, as well as their temporal, logical, and semantical dependencies. The class *concept* is the ontology's core class. Concepts are interrelated along the properties *HasPrerequisite*, *PartOf*, *Isa*, and *NececassaryPartOf* in order to describe the temporal, part of, is a, and part-whole dependencies between the various sub-concepts respectively. These properties can be employed to build authoring tools capable of defining sub-concepts at any desirable granularity level. They are also crucial to support temporal and logical navigation of the learning by providing the learner with the right learning sequence and content. In addition to the above mentioned properties, we used the class property *IsMappedTo* to annotate learning resources with ontology concepts. This property along with *HasKeyword* property, which associates keywords input by the learner to most related ontology concepts, are very useful for retrieving learning resources by mapping their metadata to ontology concepts, thus allowing resources sharing.

Learner Ontology is used to represent knowledge about the learner to deliver personalized e-learning services. This knowledge is organized into ontology concepts and relationships and used to map different contextual learner attributes onto service invocations, thus, enabling the system to discover, adapt, and deliver the most relevant learning resources in response to queries made by the learner. The main properties used in this ontology are *HasCovered* which relates individuals of class *Leaner* to domain concepts that has been covered so far, and property *ConsumedLearningResource* which relates individuals of class *Leaner* to consumed learning resources. These relationships are used to infer those concepts that have not been covered by the learner, and thus help planning his learning path. Path planning also involves the use of the domain ontology relations *HasPrerequisite*, *NececassaryPartOf*, *PartOf*, and *Isa*. Finally, the property *ConductedLearningActivity* relates the *Learner* class to *LearningActivity* class. This enables the system to infer and retrieve all previously conducted learning interactions for a particular learner. Thus knowledge embedded in activity ontology can capture all learning activities (user interactions) conducted by a learner over a period of time using a specific handheld device, as well as queries previously made by the learner. It also allows the system to recover from wireless network disconnections, which could be frequent in a mobile environment, by identifying the most recent learning activity and restoring most recent learning context. All queries made by the learner are time-stamped to infer the order in which ontology concepts were covered and their respective learning resources were consumed. This feature is crucial to organize and adjust the learning path every time a new query is made by the learner.

The device ontology however is used to represent knowledge about used devices and their hardware and software capabilities and limitations. This knowledge is very useful for the discovery of learning services whose execution profile matches the characteristics of the used device. For instance, knowledge such as maximum bandwidth that can be supported by a device; supported communication protocol; and running operating system, is needed to adapt the used device to the sensed wireless network. Other device knowledge such as enabled software applications, screen resolution, and available memory can also be used to filter out learning resources with a media type matching the device capabilities. Finally, the environment ontology formally describes the knowledge about a learner's environment which consists mainly of temporal and spatial contextual features, as well as networking, security, and connectivity issues. The main properties of this ontology are *HasLocation* which relates the class *Environment* to the current location, and the properties *HasWirelessNetwork*, *IsSecured* and *Hasbandwith* which describe the wireless network the learner is connected through, its security status, and its current bandwidth respectively. These contextual elements are very crucial to adjust learning content that is compatible, in terms of size, media-type, and privacy, with the technological set-up that characterizes the surrounding environment of the learner.

The Semantic Web Rule Language (SWRL) is used in this study to reason with the perceived context in order to retrieve metadata that can be used for the various adaptation tasks. Several SWRL rules are used to infer new context based on the sensed atomic context elements. For instance, the media-type of learning resources to be retrieved from the various distributed learning objects' repositories can be inferred from the atomic context elements describing network bandwidth, available memory and screen resolution of the used device. For example, we only search for learning objects with text type if a mobile device has a small available memory, or if the network has low bandwidth. Figure 3 describes the logical steps to select the media type of retrieved learning objects to make sure that they are browsable on the used device.

Figure 4 shows some of the experimental results for a user who want to learn about digital photography. We modeled a very simple ontology describing the digital photography subject area and asked few users to make queries related to that domain using devices with different software and hardware capabilities. The figures below show the results for a query using the concept *Camera*. The learner is provided with a list of concepts related to Camera using knowledge embedded in the domain ontology, in addition to a set of learning resources retrieved by various search agents such as Wikipedia and Youtube. Our search engine is based on mashup technology, which enables it to integrate learning resources from various distributed repositories. The retrieved resources are filtered out based on the learner, device, and environment context.

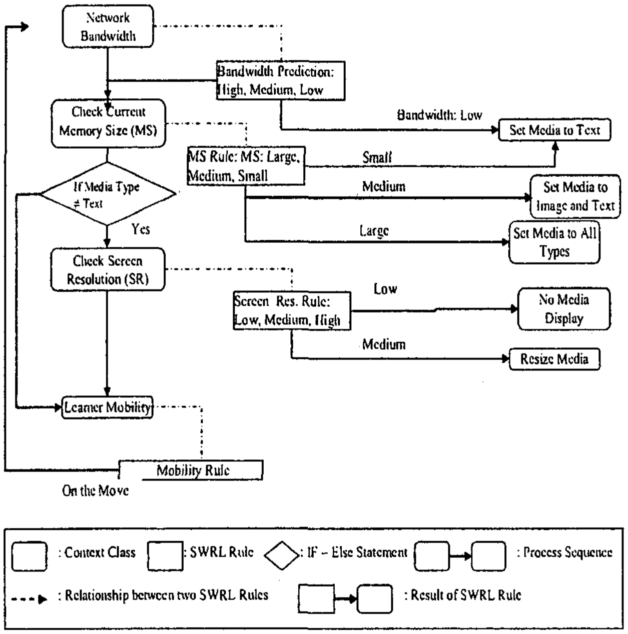


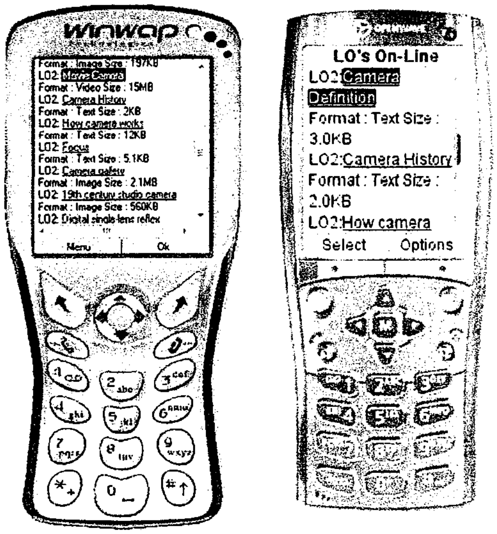
Figure 3. Media-type selection process

5. Conclusion

This paper attempts to solve some of the challenges related to context management and mobile learning design, making use of the progress made in ubiquitous computing and the Semantic Web respectively. In particular, our contribution is a method that integrates knowledge related to the learner, learning activity, used mobile technology, and surrounding environment, and defines it at the semantic level using a global interrelated ontology space. The proposed approach allows reasoning with the perceived heterogeneous context elements to translate context changes into new adaptation constraints in the operating environment, thus enabling personalized learning. An early prototype is built and the experimental results are very promising. We are currently implementing the various reasoning mechanisms to deal with the learner's context, background, and preferences.

Acknowledgement

This project is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).



Format : Image Size : 197KB
LO2: Movie Camera
Format : Video Size : 15MB
LO2: Camera History
Format : Text Size : 2KB
LO2: How camera works
Format : Text Size : 12KB
LO2: Focus
Format : Text Size : 5.1KB
LO2: Camera safety
Format : Image Size : 2.1MB
LO2: 19th century studio camera
Format : Image Size : 560KB
LO2: Digital single-lens reflex

LO's On-Line
LO2:Camera Definition
Format : Text Size : 3.0KB
LO2:Camera History
Format : Text Size : 2.0KB
LO2:How camera

Figure 4. Example of learning resources recommended to the learner.

6. References

[1] P. Bellavista, A. Corradi and R. Montanari, "A Mobile computing middleware for location and context aware Internet data services", *ACM Transactions on Internet Technology*, 6(4), 2006, pp. 356-380.

[2] V. Derballa and K. Pousttchi, "Extending knowledge management to mobile workplaces", Proc. of 6th Int. Conf. on Electronic Commerce, Delft, Netherlands, 2004, pp. 583-590.

[3] J. Davies, M. Lytras and A. Sheth, "Semantic Web Based Knowledge Management", *IEEE Internet Computing*, No. 10, 2007, pp. 14-16.

[4] S. Braun and A. Schmidt, "Don't Annoy the Informal Teacher", Proc. Int. Conf. on Knowledge Management, Graz, 2006, pp. 429-436.

[5] R. Denaux, V. Dimitrova and L. Aroyo, "Integrating Open User Modeling and Learning Content Management for the Semantic Web", Proc. 10th Int. Conf. on User Modeling, Edinburgh, UK, In *Lecture Notes in Computer Science*, Vol. 3538, 2005, pp. 9-18.

[6] P. Dolog and W. Nejdl, "Semantic Web Technologies for the Adaptive Web, I Adaptive Web: Methods and Strategies of Web Personalization", *Lecture Notes in Computer* Science, Vol. 4321, 2007, pp. 697-719.