RESEARCH   **Open Access**

Context-aware recommender for mobile learners

Rachid Benlamri[*] and Xiaoyun Zhang

* Correspondence: rbenlamr@
lakeheadu.ca
Department of Software
Engineering, Lakehead University,
955 Oliver Rd, Thunder Bay, ON P7B
5E1, Canada

Abstract

As mobile technologies become widespread, new challenges are facing the research community to develop lightweight learning services adapted to the learner's profile, context, and task at hand. This paper attempts to solve some of these challenges by proposing a knowledge-driven recommender for mobile learning on the Semantic Web. The contribution of this work is an approach for context integration and aggregation using an upper ontology space and a unified reasoning mechanism to adapt the learning sequence and the learning content based on the learner's activity, background, used technology, and surrounding environment. Whenever context change occurs, the system identifies the new contextual features and translates them into new adaptation constraints in the operating environment. The proposed system has been implemented and tested on various mobile devices. The experimental results show many learning scenarios to demonstrate the usefulness of the system in practice.

Keywords: Ubiquitous learning; Context modeling and management; Semantic web; Ontology design; Mobile learning

Introduction

The Semantic Web is an extension of the current web, whereby content is given well defined meaning so that it can be understood and processed by both software agents and humans [1]. Semantic Web provides an infrastructure with the potential to efficiently reason with content that is defined at the semantic level with formalized knowledge [2-6]. The challenge in an information-rich world is not only to make information available to people at any time, any place, and in any form, but also to offer the right content to the right person in the right way [7-9]. In particular, in the context of mobile learning, if such approach is to be adopted, many research efforts are needed to close the gap between learning strategies, knowledge discovery, and context perception and management on the Semantic Web [10]. These efforts would lead to a Semantic Web that has the potential to revolutionize the way learning services available on the web are discovered, adapted, and delivered to mobile users based on their context. To achieve this goal, we need to formally describe not only web-content, but also the various system stakeholders including content producers, content consumers, and surrounding context. This approach is adopted in this paper by integrating learner's knowledge, domain knowledge, and context knowledge. At the semantic level, meta-information and reasoning mechanisms are used to integrate these knowledge components, thus making it possible to infer the learning sequence that suits learner's profile and experience, and the learning content that suits their activity, used technology and surrounding environment.

In this paper we deal with learning in a mobile environment. Although, much progress has been made in the field of mobile services [11-15], we believe that more research is needed in the area of mobile learning. This is mainly due to the fact that most existing systems have been designed with focus on technology only [16], while they should equally address knowledge representation and learning strategies' aspects for mobile learning. In particular, several obstacles still hinder personalization of mobile learning services, such as: (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 discovery neglect contextual information on surrounding environment; and (iii) lack of context modeling and reasoning techniques that allow integration of 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 recommender system with semantic-rich awareness information. Our goal is to develop services capable of providing content recommendations tailored to the learner's background, context, and task at hand. Mobile learners expect to access such learning services from various ubiquitous setups (workplace, home, on the move, etc.) with different operational characteristics such as varying network bandwidth and limited resources on mobile devices [17]. Also, it is important to note that the operational environment of a mobile setting changes frequently. It is therefore necessary to consider a proactive context awareness mechanism that can sense both system-centric and learner-centric context and adapts the accessed services accordingly at run-time. In such setup, context aggregation can be made possible using a shared ontology space and a unified reasoning mechanism. In particular, whenever context change occurs, the system identifies the new contextual features and translates them into new adaptation constraints in the operating environment. The contribution of this paper can be summarized into three points. First, a shared ontology space for capturing, integrating and modeling contextual knowledge at a higher level based on learner context, activity context, device context and environment context. Second, a method for structuring semantic knowledge about subject-domain in such a way to enable provision of content at different granularity levels to suit learners with various background and skills. Finally, a model for run-time context management that translates context changes into system-centric and learner-centric adaptations for better personalization.

The remainder of this paper is organized as follows. Section 2 reviews related work and discusses the main challenges for developing context-aware mobile learning systems. The overall system design and architecture, including context modeling are presented in section 3. Section 4 discusses the reasoning mechanisms used to deal with context integration and management. The experimental results are discussed in section 5, and finally, conclusions are drawn and further research work is suggested.

Background and related work

Research in the field of personalized learning has been dominated by the use of ontologies and related Semantic Web technologies [18-28]. Ontology is a representation of a set of concepts within a domain and the relationships between those concepts [18]. It is used to reason about the properties of that domain. Ontology concepts are defined in terms of classes which are extended with properties. These are commonly encoded

using ontology languages such as the Web Ontology Language (OWL). In particular, ontology-based approaches have been widely used for context modeling and management [8-12]. A variety of ontologies have been adopted to model knowledge about subject domains, users, resources, and other contextual elements of the user surroundings [8,29-34]. In such setups, reasoning techniques are usually applied on metadata derived from a single ontology. However, real-life learning systems require simultaneous use of knowledge derived from multiple ontologies. Cross-ontology reasoning presents many challenges. One major challenge is related to disparity and semantic mismatching between related context elements across ontologies [35,36]. A solution to this problem is efficient integration of various ontologies into an upper ontology space capable of capturing and modeling information related to the global shared knowledge at a higher semantic level, and thus simplifying cross-ontology reasoning. Another problem of concern with ontology based approaches is their inappropriateness to reasoning with uncertainty. It should be noted that not all context elements are perceived with precision. Some of the context elements are quantized with uncertainty leading to certain ambiguity while defining and reasoning with context, [37]. This problem can be dealt with by integrating various reasoning models that may combine probabilistic, rule-based and logic reasoning techniques [38-40].

In addition to the above-mentioned problems, research in mobile learning recommenders has its own challenges. The task of a mobile learning recommender can be considered, to a certain extent, as a knowledge management task to support the learner's current activity. This is because the goal for mobile learning is not always long term learning, but, in many cases, an on-demand learning process triggered by immediate real-world needs. In such learning environment, the context of the learner matters, and the system should be able to capture the learner's profile, learner's context, task at hand, and the environment in which learning occurs. This knowledge is important to provide context-aware delivery methods that can generate content that meets the immediate learning goals. Other challenges that need to be addressed to develop robust mobile learning recommenders are those related to (i) the limited-resource mobile technology; (ii) low-bandwidth and unsecured wireless communication; (iii) heterogeneous context management and (iv) personalized content retrieval. This paper attempts to solve some of the problems related to the last two challenges, heterogeneous context management and personalized content retrieval, making use of the progress made in ubiquitous computing and the Semantic Web respectively. Before describing our system, we first overview recently published work in the field of ontology-based content retrieval and adaptation for e/m-learning. We then particularly focus on work in m-learning recommenders.

Dolog and Nejdl [33] have used knowledge extracted from four different ontologies (subject, user, resource, and link ontology) to realize personalized access to resources on the Semantic Web. Subject and resource ontologies are used to provide explicit semantic about information resources and the way they fit to user queries and goals. However, user and link ontologies are used to provide additional means for deciding which links to show, hide, and annotate according to the learner's background. The focus of their work is mainly around learner-centric adaptations. Another system that relies on learner modeling is the European Union project, Learning In Process (LIP) [41]. LIP aims to provide immediate learning content on demand for knowledge

intensive organizations through incorporating context into the design of learning systems. A matching procedure is presented to suggest personalized learning content based on user's current competency gap. The matching process tries to find relevant learning objects for a given user context by computing a similarity measure between the current user context abstraction and the ontological metadata of each learning object.

In the field of mobile learning, many approaches were adopted to develop context-aware learning services. These vary from simple systems relying solely on device content adaptation [42], to complex infrastructures for generating context aware mobile learning applications [43]. Yarandi et al.[44] have developed a mobile learning system with four modules to enable courseware management, course content adaptation, test evaluation and course recommendation mediation. The system is used to learn new natural languages. The course recommendation mediator selects suitable learning material by traversing the courseware knowledgebase according to pedagogical rules and ontological user profiles. The recommendation mediator is guided using results provided by the course-test evaluation module. Course material consists of learning objects with various granularity and their associated learning activities (writing, reading, speaking and listening). Each activity retains a learning objective, a difficulty-level, a suggested time for completion, and is associated a test score based on a probabilistic model. In a similar study, Benlamri et al. [45] have developed a combined model based on probabilistic learning techniques and ontologies to enable context processing and management in a mobile learning environment. The system uses Naïve Bayesian classifiers to deal with context elements that are quantized with uncertainty. Naïve Bayesian classifiers are used to recognize high level contexts in terms of their constituent atomic context elements involving uncertainty. Recognized contexts are then interpreted as triggers of actions yielding web service discovery and adaptation in order to achieve personalized instruction that best match the learner's needs and the operating environment. Mobiglam [46] is another mobile learning framework that is based on Bayesian techniques. Unlike the systems mentioned above, Mobiglam allows users to access resources and other learning services through Virtual Learning Environments (VLEs) such as Moodle and WebCT. Device adaptations are provided by a J2ME application installed on the client devices, while adaptation of VLE content is done on the server side through a decision engine relying on a Bayesian learning algorithm.

Unlike the systems described above, Mobilearn [43] is a complete infrastructure developed by a consortium of European universities in collaboration with MIT and Stanford to build mobile learning applications. Mobilearn is based on the Open Mobile-access Abstract Framework (OMAF). The latter is built upon two technologies – an extended version of the MIT-OKI (Open Knowledge Initiative) which provides an open and extensible architecture for learning technology [47] – and the IMS-ALF (Abstract Learning Framework) which provides an abstract representation of interoperable services and their interfaces [48]. The goal is to build a framework that can provide various specifications (i.e. various mobile applications) based on the used services, according to the open architecture approach.

Another challenge of mobile learning is its human computer interface (Mobile HCI) aspect, and especially the provision of context-aware communication and interaction. For instance, a mobile device that support a number of network adaptors, and equipped

with some sensors such a GPS (Global Positioning System) unit, a network bandwidth sensor, and a smart card for user-authentication, can provide useful context-aware functionalities that allow one to adapt the behavior of the system according to the sensed wireless network, position in space (location-awareness), unexpected network interruptions, and security threats respectively. Such system becomes able to select what content to show and how to show it based on the learner's activity and surrounding environment. An example system with such capability is the MObile PErsonal Trainer, MOPET [49]. The latter monitors users' positions and other physiological parameters in outdoor sports activities to present functionalities such as location-aware maps augmented with visualizations of users' performance, or context-aware fitness advice and 3D demonstrations of exercises. Mobile learning systems should exhibit these capabilities to achieve efficient context-aware communication and interaction.

Although our approach builds on many of the techniques described above, it differs from previous work in several aspects. First, unlike the above mentioned approaches, the proposed system tries to build a global context view out of a large set of heterogeneous contextual information, including knowledge about the learner, learning domain, learning activity, used technology, and surrounding environment. Second, unlike the systems described in [33,41,44] which use separate ontologies for reasoning with domain knowledge and context knowledge respectively, our approach uses a shared upper ontology space that permits reasoning with the entire system-centric and learner-centric contexts, thus enabling efficient context integration and adaptation. Finally, an important aspect of our approach is the ability for reasoning under uncertainty. It should be noted that not all contextual information around the learner can be quantized with certainty, yet the semantic web models are logic based with facts either being true or false. One possible solution to this problem is to simplify contextual data, which is fundamentally continuous or uncertain, down to symbolic discrete assertions. However, this process may introduce errors which can further propagate globally when repeatedly used in the reasoning process. To deal with this issue we opted for uncertainty representations like fuzzy logic which uses non-linear combination functions. Fuzzy logic uses min/max to combine values, so repeat combination of the same information does not introduce significant errors [50] in the reasoning process.

System design and architecture

The overall system architecture is presented in Figure 1. The core of the proposed system is based on a Run Time Environment (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 aggregation of the main context components as they emerge from the learner's interaction with the system. To achieve such complex task, we structured the RTE into three hierarchical levels as shown in Figure 2. 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, and users' background and preferences. The context sensing layer generates quantized and non-quantized raw data whose values are numeric values, boolean values, and literals, and most of which are time-stamped. To transform this context data into meaningful context information
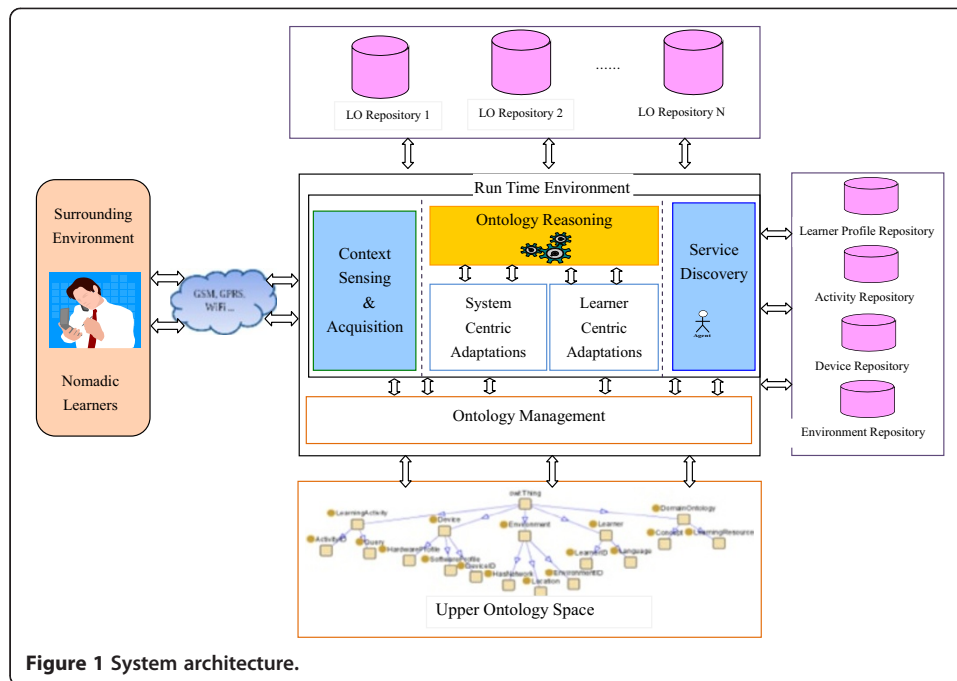
Figure 1 System architecture.

that can be used by the reasoning engine, 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, is context adaptation layer, where learning services are discovered and learning content is adapted based on perceived context. The upper ontology space describing knowledge about all context components is incremented with subject-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 learning services discovery and adaptation based on system-centric context (i.e. device and environment context) and learner-centric context (i.e. learner and activity context). In particular, the extracted metadata is used to personalize both the learning path and the learning content in order to match the learner's background, prerequisite requirements, previous tasks, available network bandwidth, network security, and other connectivity issues. Each of these adaptations is controlled by context-adaptation logic in the form of ontology reasoning.

**Context acquisition and representation**

Context is any information that can be used to characterize the situation of an entity, where an entity can be a person, a place, a physical or a computational object [51]. Context information can be classified into atomic context and composite context. Atomic context elements are associated with raw data that is either sensed or profiled. Composite context on the other hand can be derived from atomic context elements through computation, inference, or learning techniques. Example of computed composite context is user's age, or time to accomplish a learning task. Example of inferred composite context is inference of media type that can be played by a handheld device based on current network bandwidth and available memory. Finally, learned context is

Figure 2 RTE conceptual model.
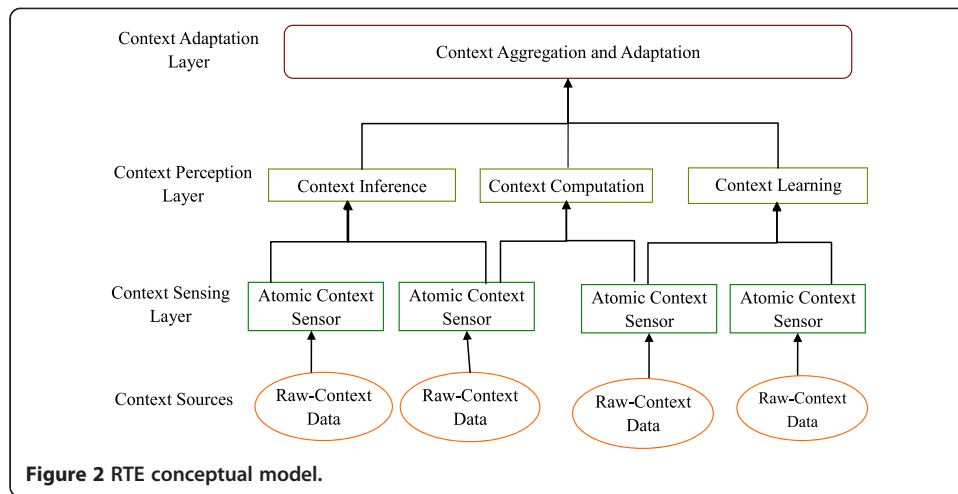
context that can be predicted from atomic context elements using prior knowledge and a probabilistic model. Example of learned context used in this study is prediction of wireless network bandwidth-range using fuzzy logic as will be shown in subsequent sections.

Many researchers have tried defining context at the higher level such as Schilit et al. [18] who decompose context into three categories – computing context – user context – and physical context. In this study, a fourth dimension of context is introduced to deal with activity context. The inclusion of activity context in a mobile setting is motivated by the fact that the needs of mobile users regarding information access are quite different from those of stationary users [52]. Usually, mobile users' needs are mainly about personalized content that is highly sensitive to their immediate environment and task at hand. Therefore, activity context will significantly improve service discovery and learning content adaptation to user needs and context.

In this study we divide context into four context groups – Learner context – Activity context – Device context – and Environment context. Figure 3 shows the way these four context categories extend each other for better context integration and service adaptation. Learner context is the main source for provisioning personalization. It also extends activity context by providing information such as learner's background, preferred language(s), and learner's schedule. Activity context however deals with accessed services; consumed learning resources; adopted learning sequences; and other subject-

Figure 3 Context integration and aggregation.
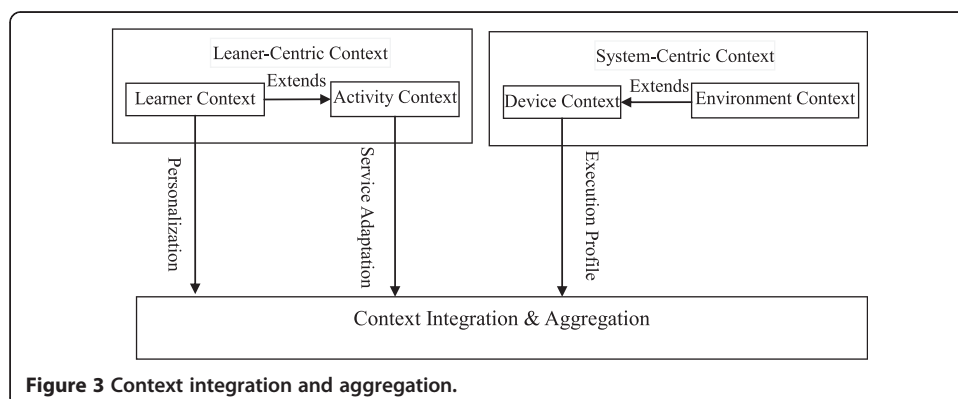
domain related knowledge. It uses subject-domain ontology as the main backbone for service adaptation and content management. Device context on the other hand is the main source for determining the software and hardware capabilities of used devices, and hence is used for setting the right execution profile for the accessed services. Information such as used operating system, screen resolution, and available memory 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 quality service features including security. Environment context extends device context by adjusting the execution profile of accessed services. For example, in determining the media type of the resources that can run on the used device at a specific point in time, we may not solely depend on the capabilities of the used device, but we should also take into consideration current network bandwidth and network security for instance.

Figure 2 shows the main components of the context acquisition system. At the lower level, a set of software and hardware services are used to sense and collect atomic context data from different sources. Whenever the learner logs into the system, the latter identifies the used device type, and then retrieves its static features from the associated device profile stored in a device profile repository (see Figure 1) which is based on the CC/PP (Composite Capabilities / Preference Profiles) standard [53]. It also senses the user's location, network channel used for connecting the learner; its connection speed (i.e. maximum theoretical speed), and the current available bandwidth [54]. It should be noted that while connection speed provides a good idea of the capabilities of the used network adapter (i.e. its upper bound limit), it does not help predict the actual network bandwidth which might be substantially less if the network is overloaded. So, ideally, the available bandwidth should be continuously monitored to report bandwidth changes as they occur. However, the process of continuously sensing and updating such dynamic context element is time and resource consuming, especially in a mobile computing environment where system resources are very limited [14]. To deal with this problem, we rather predict current network bandwidth based on previously sensed values. This is done using a fuzzy logic approach as described in section 4.1.

Table 1 shows some of the atomic context data used in this study. Each atomic context is defined by a 4-tuple (*Source_Id, Feature_Id, Absolute_Value, Symbolic_Value*). *Source_Id* refers to one of the above-mentioned four categories of context. *Feature_Id*

Table 1 Example of Atomic Context Properties

Source_ID (Type)	Feature_ID (Type)	Absolute_Value	Symbolic_Value
1 (Device)	1 (Memory Size)	Kbytes	{Small, Medium, Large}
1 (Device)	2 (Screen Res.)	(length, width)	{Low, Medium, High}
1 (Device)	3 (OS)	String	{Linux, Symbian, …}
1 (Device)	4 (Media)	String	{Text, Image, Audio, Video}
2 (Environment)	1 (Net. Bandwidth)	Kbps	{Low, Medium, High}
2 (Environment)	2 (Ne.Security)	Boolean	{Secured, NonSecured}
2 (Environment)	3 (Time)	Date-time	{Current(yyyy:mm:dd;hh:mm:ss)}
3 (Learner)	1 (Authentication Info.)	String	{Username, Password}
3 (Learner)	2 (Language)	String	{English, French, …}
4 (Activity)	1 (Query)	String	{keyword, Domain-concept}

identifies the atomic context within the context group, *Absolute_Value* is the actual value of the atomic context, and *Symbolic_Value* is the associated symbolic representation of the actual value that is used for context reasoning. The focus of this paper is on the way context data is perceived, integrated and recognized for better personalization of the learning process.

Ontology-based context modeling

At the semantic level, we define context information using an upper ontology space that includes four interrelated sub-ontologies – learner ontology – activity ontology – device ontology – and environment ontology, in addition to a domain ontology that is used to define the subject domain of interest. Thus, context aggregation is enabled using a shared ontology space and a unified reasoning mechanism across these sub-ontologies. Figure 4 shows the upper ontology space which encompasses the four context sub-ontologies and the domain ontology. Various core ontology classes describing basic ontology concepts (e.g. *Device*, *LearningResource*), role concepts (e.g. *Learner*), and role holders (e.g. *LearningActivity*), are used to interrelate concepts among the combined sub-ontologies. As shown in Figure 4, the five sub-ontologies are blended along the many properties that link various classes to form the upper ontology space. The following sections describe in details each of these sub-ontologies as well as the relationships between them.

3.2.1 Domain ontology

Domain ontology is expressed in terms of a hierarchy of subject topics, each of which is described by a set of concepts and their relationships. The class *concept* is the ontology's core class as shown in Figure 5. Concepts are related along the properties *HasPrerequisite*, *Is-a*, *HasPart*, and *HasNececassaryPart* to describe the prerequisite, is-a, part-whole, and necessary part-whole dependencies between the various concepts respectively. While these properties are important authoring tools for defining concepts' fragments (sub-concepts) at any desirable granularity level, they are also crucial to guiding learners in building their learning paths. In addition to the above mentioned
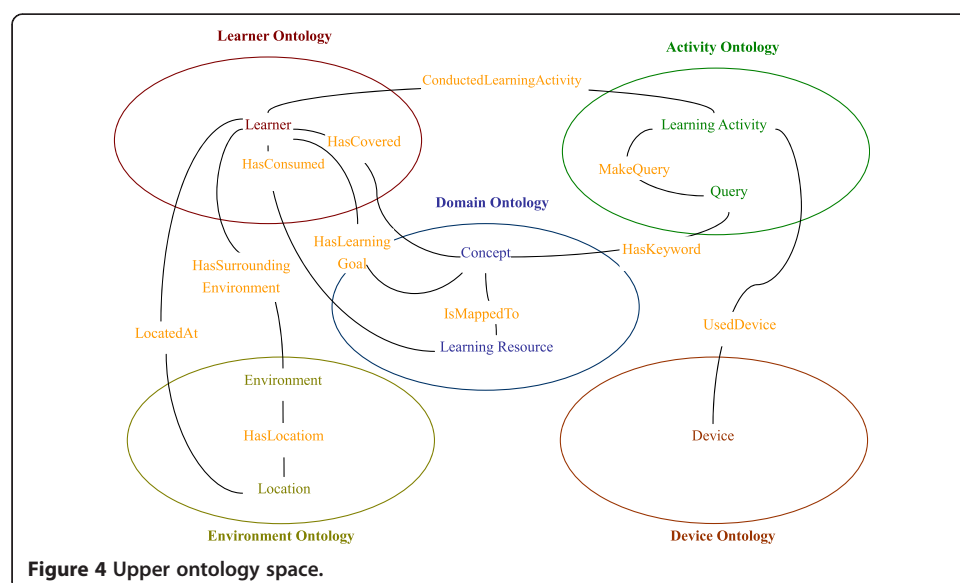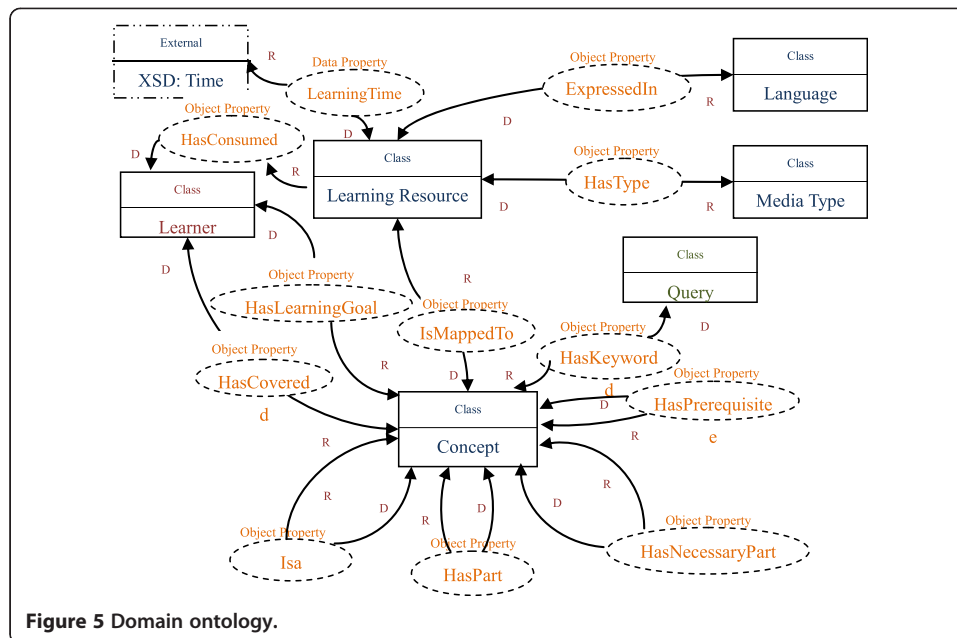

Figure 4 Upper ontology space.

Figure 5 Domain ontology.

properties, we use the class property *IsMappedTo* to annotate learning resources with ontology concepts. This property along with *HasKeyword* and *HasLearningGoal* properties, which associate keywords input by the learner to most related ontology concepts and learning goals, are very useful for retrieving learning resources by mapping their metadata to ontology concepts, thus allowing resource sharing. Another important property is *LearningTime* which associates learning resources with the typical learning time it takes a learner to consume them. This feature enables time-constrained learning by restricting the search for those learning resources that fall within the learner's allotted time. Time constrained learning is crucial in mobile learning which is time-dependent in nature. In this study, we opted for the IEEE LOM (Learning Object Metadata) standard [55] to represent learning resources. LOM introduces a base schema that abstracts data elements for learning resources into metadata with nine categories capable of describing content granularity, learning time, and semantic relationship between described concepts.
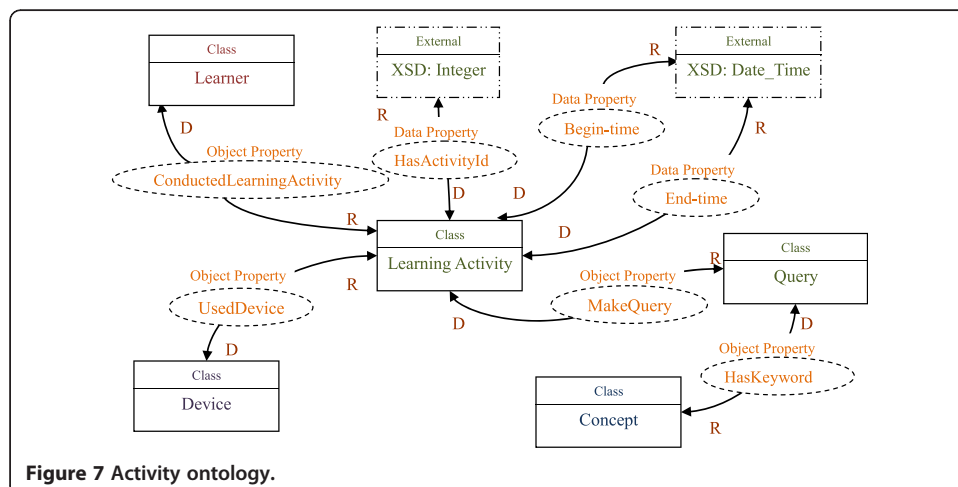
3.2.2 Learner ontology

Learner ontology is used to capture knowledge about the learner, thus enabling the system to discover, adapt, and deliver the most relevant learning resources based on their needs, background, and preferences. As shown in Figure 6, learner ontology is related to domain ontology along *HasCovered* and *HasConsumed* properties which link learners to covered concepts and consumed learning resources respectively. In particular, when a new learning path is constructed, all consumed learning resources are automatically discovered and removed for better personalization, and further learning resources are suggested to fulfill the learning goal. The latter is captured through the *HasLearningGoal* property which associates learners to one or more concepts they are interested in. Finally, the property *ConductedLearningActivity* is used to track previously conducted learning interactions by a particular learner. Other properties are also used to determine learner's location, preferred language(s), surrounding environment, and available learning time.

Figure 6 Learner ontology.
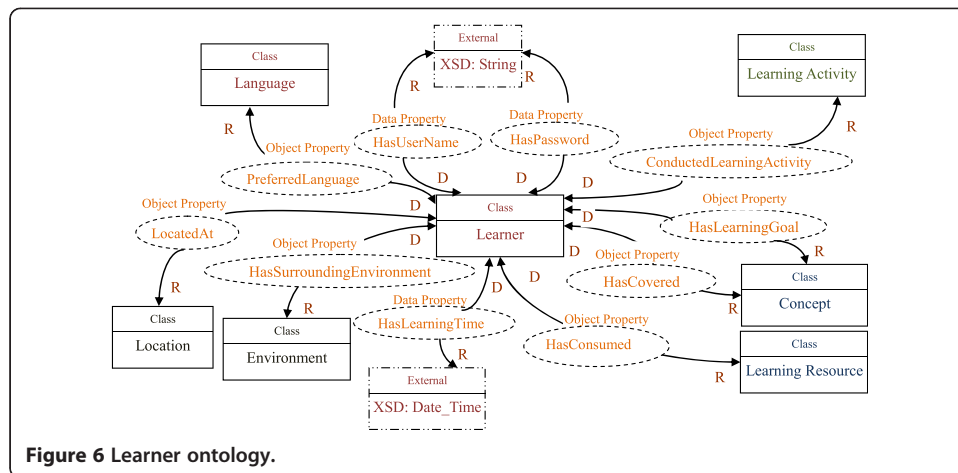
3.2.3 Activity ontology

Activity ontology is used to capture knowledge about a learning activity (i.e. set of user interactions) conducted by a learner over a period of time using a specific handheld device. Knowledge embedded in this ontology allows the system to infer history of all user interactions with the system, including queries made by the learner and used devices. 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. Figure 7 shows the main concepts used in activity ontology along with their relationships with other ontology concepts. The property *UsedDevice* links learning activities to devices that were used to conduct them. Device is deliberately linked to learning activity as learners may use different devices in their various interactions with the system. The property *MakeQuery* allows inferring all queries made during a learning interaction with the system. Queries are linked to concepts along the *HasKeyword* property. The system is designed to help learners use ontology vocabulary to compose their queries. The data property *QueryTime* is used to time stamp all queries made by the learner in order to infer the order in which ontology concepts were covered and their respective learning resources were consumed. This important temporal feature is crucial to organize and adjust the learning path

Figure 7 Activity ontology.

every time a new query is made or a new learning resource is invoked by the learner. Information such as begin-time and end-time of a learning activity can also be retrieved from the ontology to track the most recent interaction with the system, and to deal with the learner's available time.

3.2.4 Device ontology

Device ontology is used to capture knowledge about used devices and their hardware and software capabilities as well as their limitations. This knowledge is very useful for discovery of learning services whose execution profile matches the characteristics of the used device. Our device ontology is conforming to the CC/PP standard [53]. The latter is an industry standard for describing a delivery context, a set of attributes that characterizes the capabilities of the access mechanisms and the preferences of the user. Using CC/PP, allows separating content-design from content-delivery. CC/PP represents device capabilities using two-level hierarchy consisting of hardware and software attributes. As shown in Figure 8, the device ontology is based on the CC/PP components, *HasSoftwareProfile* and *HasHardwareProfile*. These two ontology classes define knowledge related to the software-centric and hardware-centric features of a device. For instance, information about device operating system, supported applications, and screen resolution are essential to discover learning services whose software and hardware requirements match device capabilities. Other properties embedded in the device ontology can also be used for personalization. For example, the property *SupportedLanguage* is used to infer whether the used device supports a language preferred by the learner. Learner ontology and device ontology are linked along property *UsedDevice* which enables inferring which device is used by a particular learner during a specific learning activity.

3.2.5 Environment ontology

Environment ontology formally describes the knowledge about a learner environment which consists mainly of temporal and spatial contextual features, as well as networking, security, and connectivity issues. Figure 9 shows the main ontology classes and their relationships. The property *HasLocation* links the Environment class to the current learner's location, while the properties *HasWirelessNetwork*, *IsSecured* and *HasBandwidth* describes the wireless Network the learner is connected to, its security status, and its current bandwidth. These contextual elements are crucial to infer and
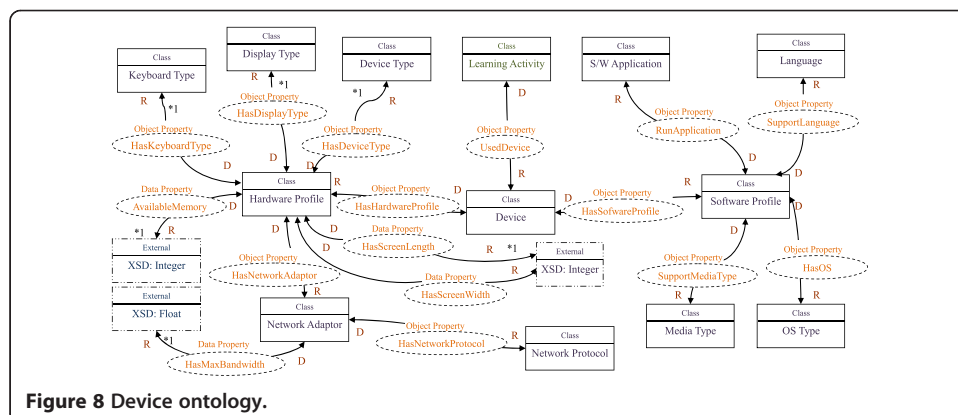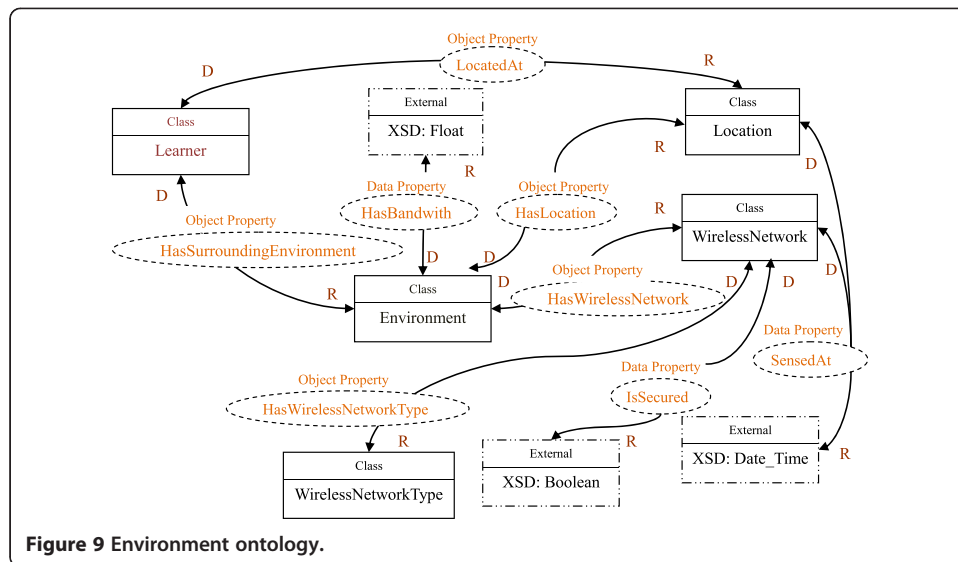

Figure 8 Device ontology.

Figure 9 Environment ontology.

adjust learning content that is compatible, in terms of size, media-type, and security with the technological set-up that characterizes the surrounding environment.

Ontology reasoning

Understanding the importance and the role of tacit knowledge is key element to formulating knowledge management strategies for mobile-learning systems. Unlike explicit knowledge that is coded knowledge, tacit knowledge is knowledge that is embedded in system actors (i.e. learners and other dynamic context elements) [56] and which is difficult to be shared and distributed. In this study, the upper ontology space, built out of the five integrated sub-ontologies is used in such a way to allow tacit knowledge related to learners and their context to be defined at the semantic level and used to a great extent in the personalization process. In particular, knowledge in the upper ontology space is structured in such a way to allow building a conceptual learner model out of a sequence of learning activities by linking a user's conceptualization to particular subject domain ontology. This constitutes the key design aspect of our upper ontology space to enable personalized learning. A learning activity is characterized by user interactions, through which contextual information related to the user's surroundings, as well as concepts covered, queries performed, and learning resources consumed are stored and used as knowledge facts to enable further inferences aiming to achieve better adaptation. Thus, the perceived context and the accessed domain information are efficiently used for building a personalized learning path that is aware of the learner's interaction history, preferences, background knowledge, and operating environment. The used approach models the learner at the semantic level by providing a formal learner's conceptualization defined in OWL, and thus allows reasoning upon it in order to infer the learner's understanding of the subject domain. The reasoning is performed in terms of SWRL (Semantic Web Rule Language) rules that are applied on knowledge represented in the OWL-DL (Description Logics) ontology. It should be noted here that reasoning in systems integrating DL ontologies and rules is a very hard task [56]. This is mainly due to undecidability of reasoning in such systems. This is particularly the case

for those systems integrating DL ontologies with recursive or hybrid rules [57,58]. So, bridging the discrepancy in these two knowledge representations is a challenging problem. Many studies [57-60] have shown that to avoid undecidability of reasoning, practically all decidable approaches to integrating ontologies and rules should impose specific conditions which restrict the interaction between the rules and the ontology. In this work, we adopt a similar decidable reasoning approach by adhering to restrictions, such as avoiding recursive and hybrid rules. In particular, we use the SWRL-Jess Bridge, a bridge that provides the infrastructure for incorporating the Jess rule engine into Protégé-OWL to execute SWRL rules. The system also relies on some SWRL-built-in libraries, which include an implementation of the core SWRL built-ins, as well as mathematical built-ins, to support the use of complex expressions in rules, reasoning with temporal information and querying OLW ontologies.

The sequence of steps given below illustrates the personalization process adopted by our system in a typical learning scenario where a learner submits a query in a specific subject domain area and receives a planned learning sequence fulfilling the learning goal. This learning scenario is also depicted graphically in Figure 10.

1. When the learner logs in, his background, preferences, and previous learning activity are retrieved.
2. The learner uses the domain ontology vocabulary to query the system.
3. The subject-domain ontology related to the learner's query is identified and retrieved.
4. Based on the learner's query, the system infers the related ontology concept(s) and identifies those concepts that are part of prerequisite knowledge, core-knowledge, and related knowledge using *HasPrerequisite*, *HasNecessaryPart*, and *HasPart* 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.
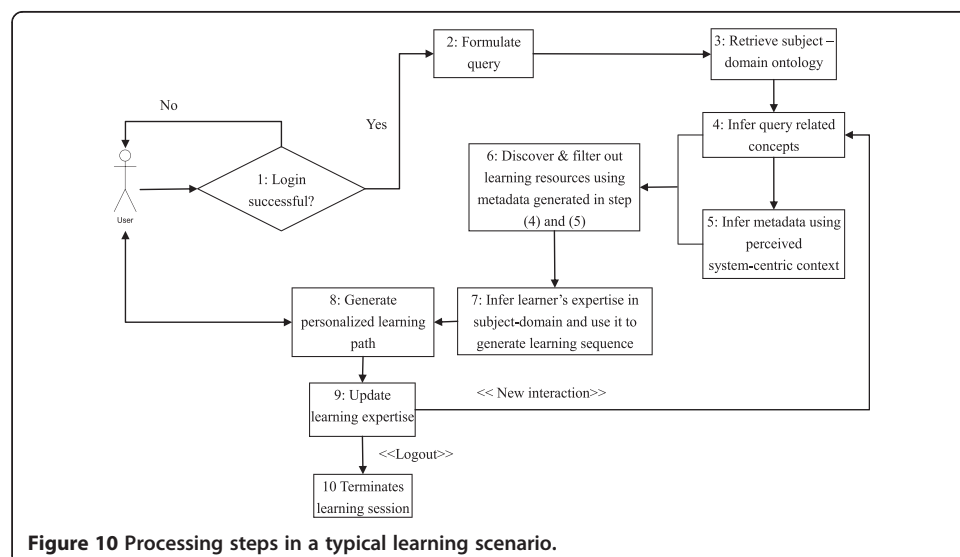
**Figure 10 Processing steps in a typical learning scenario.**

6. The metadata generated in (4) dealing with related domain ontology concepts, and the system-centric metadata generated in (5) are then used to discover and filter out learning resources from various learning repositories.

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 that is aware of the learner's history and available learning time. Thus, the newly constructed learning sequence consists of optimized system-centric learning resources to fulfill the current learner's activity and goals.

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 the system, or when all domain concepts related to current activity have been covered.

The above strategy strives to meet the best personalized learning path by dynamically updating the learning sequence based on the learner interactions with the system and surrounding environment. Below, we give a detailed description of the various system-centric adaptations and learner-centric adaptations used in the personalization process. We also provide the algorithm used to help navigate a learning path.

4.1 System-centric adaptation

In a mobile environment it is important for the system to consider system-centric adaptations to cope with the lack of resources in used devices and the unsecured low-bandwidth wireless network. Thus, system-centric adaptations aim at filtering out those learning resources that can efficiently be transmitted over the network and properly run on used devices. These adaptations are triggered by the context monitoring process which identifies context changes and proactively performs actions such as restricting media type and pruning large learning resources from the learning path in case of low network bandwidth. A number of inference rules have been developed to operate on perceived device and environment atomic context elements to achieve this goal. The diagram shown in Figure 11 describes the logical steps used to achieve the main system-centric adaptations considered in this study.

When the learner logs into the system, first, the system senses the used network adaptor and retrieves its connection speed. Connection speed is an attribute that is straightforward to obtain and it represents the maximum theoretical speed for the used wireless adapter [56]. Knowing the type of the network connection, such as IEEE 802.11 or GPRS, gives our reasoning engine the insight that allows it to make some adaptation choices related to media-type and size of resources that are 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 does not need to sense the network bandwidth and does not need to make any restrictions on media type because the available bandwidth is stable and large enough to handle all type of resources. However, if the sensed connection is GPRS, the system adapts the media-type and size based on the available bandwidth as explained below. For example, for a GPRS
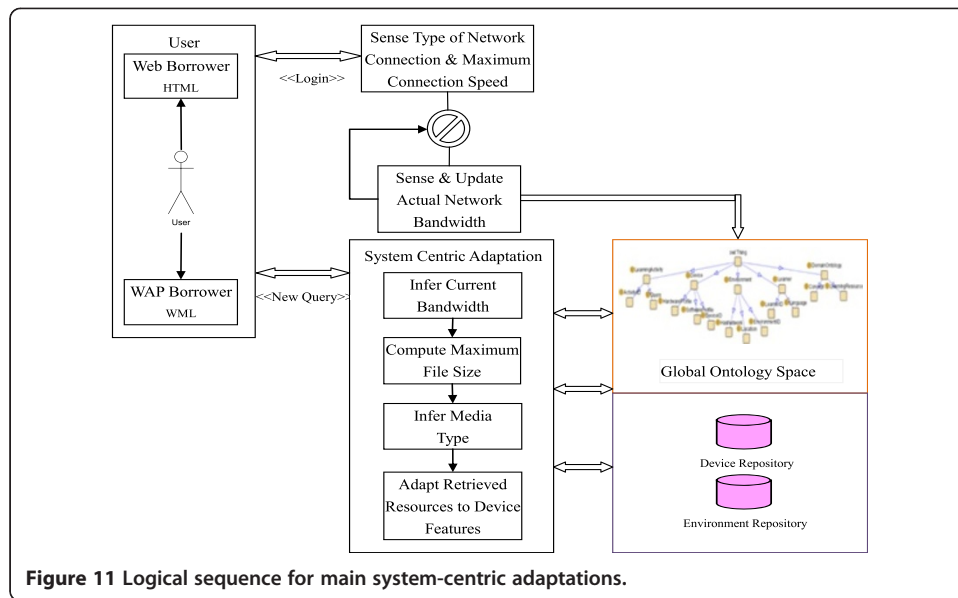
Figure 11 Logical sequence for main system-centric adaptations.

connection with a maximum connection speed of 48 kbps, the actual network bandwidth is usually less than 48 kbps due to traffic on the network [54]. Ideally, the system should continuously sense and update the current network bandwidth whenever bandwidth change occurs. However, the process of continuously sensing and updating the ever changing bandwidth is time and resource consuming as it involves sending data packets through the network. To solve this problem, we only sense the actual bandwidth at some points in time, and we use a fuzzy logic approach in conjunction with SWRL rules to predict the available bandwidth between these points. Also, to reason with bandwidth, we translate the predicted current bandwidth into meaningful symbolic values such as *low*, *medium*, and *high* bandwidth. Fuzzy logic is also used to predict the maximum size of learning resources that can be communicated without incurring long delays. For instance, we only search for learning resources with text type if a mobile device, operating on a GPRS network for instance, has very low bandwidth. However, we can extend media type to image and video if the network bandwidth is high and the available device memory is large. We also perform other checks to adapt to features such as screen resolution, used operating system, and network security. Below we describe the fuzzy logic approach used to predict media type and size and we give a full scenario to illustrate all system-centric adaptations.

We make use of the fuzzy logic truth values in conjunction with SWRL rules to allocate symbolic value to the predicted current network bandwidth. Figure 12 shows the membership function for network bandwidth. This is used to predict the network bandwidth using the fuzzy qualifying linguistic variables such as *low*, *medium*, and *high*. Note that *Maxband* which stands for maximum bandwidth is associated with the maximum connection speed of the used wireless adaptor. The symbol $\mu_A(x)$ represents a truth value that is between 0 and 1. Based on the membership function given in Figure 12, $\mu_A(x)$ can be computed by (1).
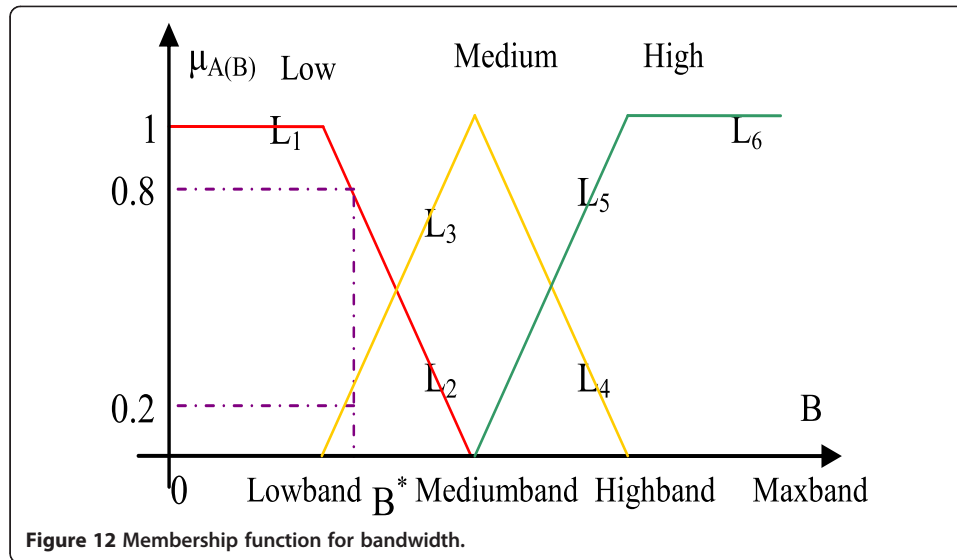
Figure 12 Membership function for bandwidth.

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

In Rule-Set-1 we describe the SWRL rules that are used to infer the truth values of classified symbolic network bandwidth given in (1). For the sake of space we only give *TruthValueRule1* and *TruthValueRule2* which are respectively related to L_1 and L_2 of (1), and which are used to infer the truth values associated to low network bandwidth. The abstract SWRL syntax which is consistent with the OWL specification is rather verbose and not particularly easy to read [61]. Instead, we use a relatively informal "human readable" form where both antecedent and consequent are conjunctions of atoms with variables prefixed with a question mark (e.g., ?x), and which may also include functional notations as shown in Rule-Set-1. The latter shows the two above-mentioned rules which use the property *UsedDevice(?act,?dev)* which associates a learner identified by an activity identifier *act* to their handheld device *dev*, and the data properties *HasBandwidth* and *MaxBandwidth* which describe respectively the current network bandwidth and maximum connection speed of a used handheld device.

Rule-Set-1 SWRL rules for fuzzy logic truth values
TruthValueRule1
ActivityID(?act) ∧ UsedDevice(?act, ?dev)∧ HasBandwidth(?dev, ?b) ∧ HasNetworkA-daptor(?dev, GPRS) ∧MaxBandwidth(?dev,?Maxband) ∧ swrlb:multiply(?Lowband,? Maxband, 0.25) ∧ swrlb:lessThanOrEqual(?b, ?Lowband)→ProbLow(?dev, 1.0) ∧ NetworkBandwidth(?dev, "Low").

TruthValueRule2

ActivityID(?act) ∧ UsedDevice(?act, ?dev) ∧ HasBandwidth(?dev, ?b) ∧ HasNetworkA-daptor(?dev, GPRS) ∧ MaxBandwidth(?dev,?Maxband) ∧ swrlb:multiply(?Lowband,? Maxband,0.25) ∧ swrlb:multiply(?Mediumband, ?Maxband, 0.5) ∧ swrlb:greaterThan(?b, ?Lowband) ∧ swrlb:lessThanOrEqual(?b, ?Mediumband) ∧ swrlb:subtract (?z1, ?Med-iumband, ?b) ∧ swrlb:subtract(?z2, ?Mediumband,?Lowband) ∧ swrlb:divide(?z, ?z1, ? z2)→ProbLow (?dev, ?z) ∧ NetworkBandwidth(?dev, "Low")

The following real-life scenario illustrates the way we apply the SWRL rules shown in Rule-Set-1. For instance, let's assume that *Irene* is using the "Motorola W270" mobile device model, called here *MotoW270* which supports a maximum connection speed of 32.0 kbps. Let's also assume that the value (x^*) which represents the most recently sensed network bandwidth is found to be around 18.0 kbps. This is fluctuating between medium to high bandwidth with respect to the maximum connection speed as shown in Figure 12. When *TruthValueRule4* and *TruthValueRule5* are applied, facts (*A2*) and (*B2*) are inferred, resulting into the addition of four statements to the list of facts as shown in Rule-Set-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.

The inferred probabilities associated with current bandwidth are then used to determine the maximum allowable resource size. The aim here is to target resources which have reasonable size, as it is not practical to offer the learner a large learning resource (e.g. few Mbytes) if the used device operates on a low bandwidth (e.g. few kbps). So, we conducted few experiments on three devices with different capabilities to identify some typical threshold values related to resource sizes that can be used for specific bandwidth ranges. The used devices are a basic Nokia phone emulator, a Sony-Ericson W830C and an HTC-S261 smartphone which have a maximum bandwidth of 32kpbs, 48 kbps and 120 kbps respectively. We used these devices to request learning resources of different media type and size under a network bandwidth varying from 8 kbps to 120 kbps. Based on these experiments, we have adopted the following assumptions aiming to keep a reasonable response time. If a mobile device has a connection speed less than 32 kbps, we should not consider resources that exceed 500Kbytes. However, if the connection speed was between 32 kbps to 66 kbps, then resources over 1Mbytes should not be considered. We used these threshold values along with the previously inferred symbolic values associated with current network bandwidth to predict the maximum allowable resource size.

Rule-Set-2 Examples of truth value inferences
Ontology related facts

A1) before applying TruthValueRule4

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)

B1) before applying TruthValueRule5

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:lessThanOr Equal(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)

OInferred facts

A2) after applying TruthValueRule4

ProbMedium(MotoW270, 0.75) NetworkBandwidth(MotoW270, "Medium")

B2) after applying TruthValueRule5

ProbHigh(MotoW270, 0.25) NetworkBandwidth (MotoW270, "High")

It should be noted that the system is designed in such a way that the maximum tolerable response time, that depends on the user's activity and the nature of requested resources, can be easily modified to accommodate learners with more or less restrictive time constraints. Figure 13 shows the main components of the used fuzzy system which consists of singleton fuzzifier, product inference engine, fuzzy rule base, and center average defuzzifier. The system starts with a fuzzification of the input variable, then rule evaluation, followed by aggregation. The latter is the process of unification of the outputs of all rules. The last step of the fuzzy system is defuzzification to obtain a crisp output [50]. Figure 14 shows the membership function for resource 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 resource size. We also define three fuzzy sets *Small*, *Medium*, and *Large* as the output space (resource size) as shown in Figure 14. Note that we use *SmallSize*, *MediumSize*, and *LargeSize* to refer to center average values for *small, medium, and large* fuzzy sets respectively. The fuzzy rule base consists of three simple rules as shown below.

R^1: if network bandwidth *(B)* is *Low* then resource size *(Z)* is set to *Small*
R^2: if network bandwidth *(B)* is *Medium* then resource size *(Z)* is set to *Medium*
R^3: if network bandwidth *(B)* is *High* then resource size *(Z)* is set to *High*
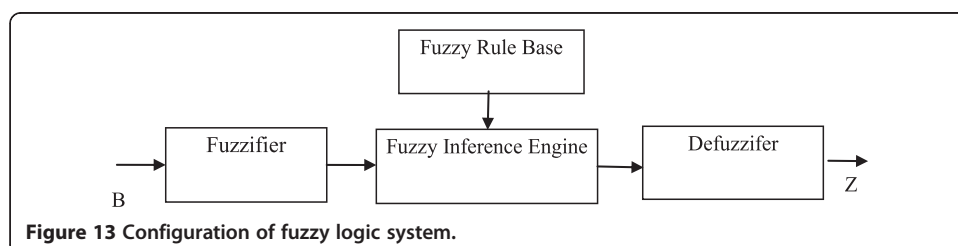
Figure 13 Configuration of fuzzy logic system.

Figure 14 Membership function for resource size.
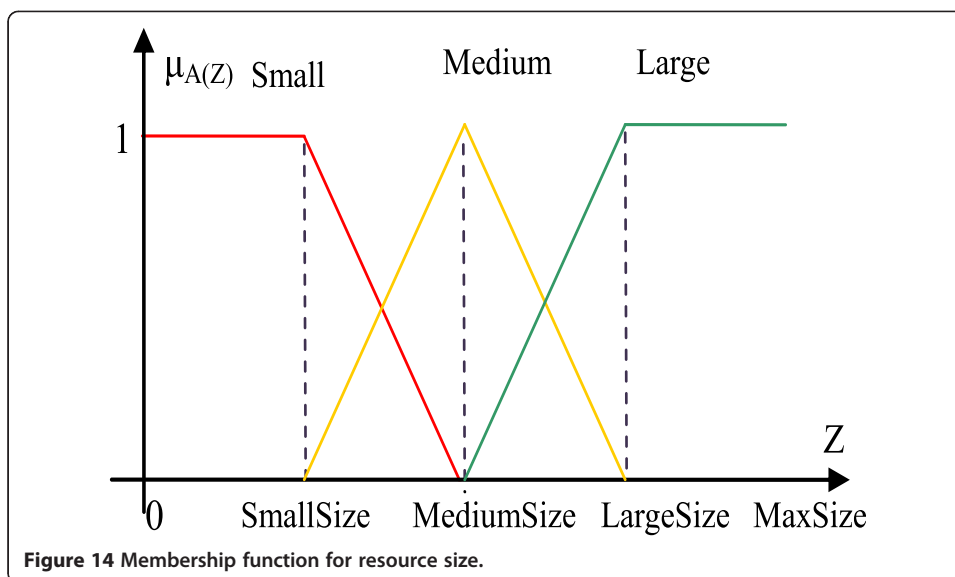
Suppose that the fuzzy set F^k in the fuzzy rule base R^k is normal with center $\bar{y}^k$. Then the crisp output from the fuzzy system with singleton fuzzifier, product inference engine, center average defuzzifier, and rule base R^k where, R^k is defined as follows: **if** x_1 is A_1^k and... and x_n is A_n^k , **then** y is F^k, $k = 1,...N$ is given by [50]:

$$y^* = \frac{\sum_{k=1}^{N} \bar{y}^k \left(\prod_{i=1}^{n} \mu_{A_i^k}(x_i^*) \right)}{\sum_{k=1}^{N} \left(\prod_{i=1}^{n} \mu_{A_i^k}(x_i^*) \right)} \tag{2}$$

In our case, (2) results into:

$$Z^* = \frac{\sum_{i=1}^{3} \bar{Z}_i * \mu_{A^i}(B^*)}{\sum_{i=1}^{3} \mu_{A^i}(B^*)}$$

$$= \frac{SmallSize * \mu_{Low}(B^*) + MediumSize * \mu_{Medium}(B^*) + LargeSize * \mu_{Large}(B^*)}{\mu_{Low}(B^*) + \mu_{Medium}(B^*) + \mu_{Large}(B^*)} \tag{3}$$

In Rule-Set-3, we show the SWRL rule (*ResourceSizeRule*) associated with equation (3). In *ResourceSizeRule*, the data properties *ProbLow*, *ProbMedium*, and *ProbHigh* are those probabilities obtained in *TruthValueRules* (see Rule-Set-1). To show how these rules are applied in our system, we provide a real-life scenario. Let's assume that our learner *Irene* is using a GPRS connection with a maximum connection speed of 32 kbps. This connection speed delimits the maximum resource size to 500Kbytes as described above. These assumptions are represented by facts (*A1*) in Rule-Set-3. When *Resource SizeRule* is applied, fact (*A2*) is inferred resulting into the addition of statement *Resource Size(MotoW270*, 281.25) to the list of facts. Indeed, since the previously sensed

network bandwidth was 18 kbps, our system chooses not to exchange resources over 281.25Kbytes as deduced from the set of inferences shown in Rule-Set-3.

Rule-Set-4 describes the SWRL rules used to select the media type of retrieved learning resources based on predicted current bandwidth. The data properties *Network Bandwidth* and *AvailableMemory* represent respectively the current bandwidth and available device memory. In *MediaRule1-to-3*, the system sets the media type to the appropriate format (i.e. text, image, video) based on the value of predicted current network bandwidth. The system also sets the maximum allowable resource size, as computed in Rule-Set-3, based on the device's available memory. If the size of the device memory is smaller than the maximum allowable media size computed in Rule-Set-3, then *AllowedResourceSizeRule1* sets the maximum media size to the device memory size; otherwise the maximum media size remains unchanged as stated in *AllowedResourceSizeRule2*.

Rule-Set-3 SWRL rule and inferences for determining maximum allowed resource size
ResourceSizeRule

ActivityID(?act) ∧ UsedDevice(?act, ?dev) ∧ ProbLow(?dev, ?Tl) ∧ ProbMedium(?dev,?Tm) ∧ ProbHigh(?dev, ?Th) ∧ MaxSize(?dev, ?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(?dev, ?z)

Ontology related facts

A1) before applying *ResourceSizeRule*

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)

Inferred facts

A2) after applying *ResourceSizeRule*

FileSize (MotoW270, 281.25)

To show how the above rules are applied we use the previous scenario of learner *Irene* who is using device *MotoW270* operating at a bandwidth of 18.0 kbps to access

the system services. Following the reasoning shown in Rule-Set-2, the system infers a bandwidth fluctuating between medium to high as shown in facts (*A1*) and (*B1*). When applying *MediatypeRule2* and *MediatypeRule3*, facts (*A2*) and (*B2*) are respectively inferred and added to the list of facts as shown in Rule-Set-5. In (*C1*), *AllowedResource SizeRule2* is applied to compare the maximum allowed resource size previously inferred in Rule-Set-3 with the device's available memory, leading to the addition of statement *AllowedSize*(*MotoW270*, 281.25) to the list of inferred facts as shown in (*C2*). The outcome of the ontology reasoning for this scenario is that all types of media can be selected for delivery, while their size should not exceed 281.25Kbytes for them to be efficiently ported on to the used device, and thus avoiding long communication delays.

Rule-Set-4 SWRL rules for media type and resource size selection
MediaRule1

ActivityID(?act) ∧ UsedDevice(?act,?dev) ∧ NetworkBandwidth(?dev,"Low") →HasMediaType(?dev, Text)

MediaRule2

ActivityID(?act) ∧ UsedDevice(?act, ?dev) ∧ NetworkBandwidth(?dev,"Medium") → HasMediaType(?dev, Text) ∧ HasMediaType(?dev, Image)

MediaRule3

ActivityID(?act) ∧ UsedDevice(?act, ?dev) ∧ NetworkBandwidth(?dev, "High") → HasMediaType(?dev,Text) ∧ HasMediaType(?dev,Image) ∧ HasMediaType(?dev,Video)

AllowedResourceSizeRule1

ActivityID(?act) ∧ UsedDevice(?act, ?dev) ∧ FileSize(?dev,?Size) ∧ AvailableMemory(?dev,?MemorySize) ∧ swrlb:lessThan(?MemorySize,?Size) → AllowedSize(?dev,?MemorySize)

AllowedResourceSizeRule2

ActivityID(?act) ∧ UsedDevice(?act,?dev) ∧ FileSize(?dev,?Size) ∧ AvailableMemory(?dev, ?MemorySize) ∧ swrlb:greaterThanOrEqual(?MemorySize, ?Size) → AllowedSize(?dev,?Size)

Rule-Set-5 Inferences for media type selection
Ontology related facts

A1) before applying MediatypeRule2

ActivityID(Irene)
UsedDevice(Irene, MotoW270) NetworkBandwidth(MotoW270,"Large")

B1) before applying MediatypeRule3

ActivityID(Irene)
UsedDevice(Irene, MotoW270) NetworkBandwidth(MotoW270,"Large")

C1) before applying AllowedResourceSizeRule2

ActivityID(Irene)

UsedDevice(Irene, MotoW270) FileSize(MotoW270, 281.25) AvailableMemory(MotoW270, 1024.0) swrlb:greaterThanOrEqual(1024.0, 281.25)

Inferred facts

A2) after applying MediatypeRule2

HasMediaType(MotoW270,Text)
HasMediaType(MotoW270,Image)

B2) after applying MediatypeRule3

HasMediaType(MotoW270,Text)
HasMediaType(MotoW270,Image)
HasMediaType(MotoW270,Video)

C2) after applying AllowedResourceSizeRule2

AllowedSize(MotoW270,281.25)

Another type of adaptation considered in this study is language adaptation that takes into account the language preferred and used by the learner. *LanguageRule* in Rule-Set-6 establishes a constraint represented by the relationship *SearchLanguage* that associates a language with a specific learner activity. The property *PreferredLanguage*(?*act*, ?*lan*) links an activity identifier *act* to a preferred language *lan*. The property *Support Language*(?*dev*, ?*lan*) links learner's handheld device *dev* to its support language *lan*. 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 *LanguageRule1*, as shown by (*A1*) in Rule-Set-6, we can infer (*A2*) that is *SearchLanguage* (*Irene*, *French*), confirming that French can be used as a search language because it is supported by the used device. The knowledge base developed in this study also includes rules that deal with other system-centric adaptations such as network security, operating system compatibility, and screen resolution.

Rule-Set-6 SWRL rule for language adaptation
LanguageRule
ActivityID(?act) ∧ UsedDevice(?act,?dev) ∧ PreferredLanguage(?act, ?lan) ∧ SupportLanguage(?dev, ?lan) → SearchLanguage(?act, ?lan)

Ontology related facts

A1) before applying *LanguageRule*

ActivityID(Irene)
UsedDevice(Irene, MotoW270)
PreferredLanguage(Irene, French) SupportLanguage(MotoW270, English) Support Language(MotoW270, French)

A2) after applying *LanguageRule*

SearchLanguage(Irene, French)

Rule-Set-7 SWRL rules for learning sequence construction

***SimilarResourceRule1*: ConductedLearningActivity(?L,?act) ∧ MakeQuery(?act,?Q) ∧**
HasKeyword(?Q,?C) ∧ IsMappedTo(?C,?LR) ∧ HasLearningTime(?L, t) ∧
LearningTime(?LR, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → SimilarLR(?act,?LR)

***SimilarResourceRule2*: ConductedLearningActivity(?L,?act) ∧ MakeQuery(?act,?Q) ∧
HasKeyword(?Q,?C) ∧ Has(?C,?Ci) ∧ ¬Covered(?L,?Ci) ∧ IsMappedTo(?Ci,?LRi) ∧
¬Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:
greaterThanOrEqual(?t, ?t1) → SimilarLR(?act,?LRi)

***SimilarResourceRule3*: ConductedLearningActivity(?L,?act) ∧ MakeQuery(?act,?Q) ∧
HasKeyword(?Q,?C)∧ Isa(?C,?Ci)∧¬ Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧ ¬Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → SimilarLR(?act,?LRi)

***PrerequisiteResourceRule:* ConductedLearningActivity(?L,?act)∧MakeQuery(?act,?Q)∧
HasKeyword(?Q,?C)∧ HasPrerequisite(?Q,?Ci)∧ ¬ Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi) ∧¬Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → PrerequisiteLR(?act,?LRi)

***CoreResourceRule1:* ConductedLearningActivity(?L,?a) ∧ MakeQuery(?act,?Q) ∧
HasKeyword(?Q,?C)∧ HasNecessaryPart(?Q,?Ci) ∧ ¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧ ¬ Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → CoreLR(?act,?LRi)

***CoreResourceRule2:* ConductedLearningActivity(?L,?a) ∧ MakeQuery(?act,?Q) ∧
HasKeyword(?Q,?C) ∧ IsNecessaryPartOf(?Q,?Ci) ∧ ¬Covered(?L,?Ci) ∧ IsMappedTo(?Ci,?LRi) ∧ ¬ Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → CoreLR(?act,?LRi)

***NonCoreRelatedResourceRule1:* ConductedLearningActivity(?L,?act) ∧ MakeQuery(?act,?Q) ∧
HasKeyword(?Q,?C) ∧ HasPart(?Q,?Ci) ∧ ¬Covered(?L,?Ci) ∧ IsMappedTo(?Ci,?LRi) ∧ ¬Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → NonCoreRelatedLR(?act,?LRi)

***NonCoreRelatedResourceRule2:* ConductedLearningActivity(?L,?act)∧MakeQuery(?act,?Q)∧HasKeyword(?Q,?C) ∧ IsPartOf(?Q,?Ci)∧ ¬Covered(?L,?Ci)∧ IsMappedTo(?Ci,?LRi)∧ ¬Consumed(?L,?LRi) ∧ HasLearningTime(?L, t) ∧ LearningTime(?LRi, ?t1) ∧ swrlb:greaterThanOrEqual(?t, ?t1) → NonCoreRelatedLR(?act,?LRi)

4.2 Learner-centric adaptation

The best personalization of learning is the one that supplements domain knowledge with the learner's tacit knowledge. This approach is adopted in the learner-centric adaptation process to produce ontology-compliant learning sequence aware of the learner's background and task at hand. As shown in Figure 10, first, the system infers concepts related to the learner's query using knowledge embedded in the subject domain ontology. This is done to ensure that the initial learning sequence is ontology compliant.

Then, knowledge related to learner's context is used to avoid reiterated covered concepts and consumed learning resources. In Rule-Set-7, we show the rules used to derive the initial learning sequence. First, the reasoning engine uses *Similar ResourceRule1-to-3* to check for resources that are directly related to concept(s) in the learner's query. Second, *PrerequisiteResourceRule* is applied to infer prerequisite knowledge based on *HasPrerequisite* relationship. Third, the reasoner checks for core knowledge using *CoreResourceRule* which is based on the necessary part-whole (*HasNecessaryPart*) relationship, thus inferring sub-concepts that should be covered to completely understand the queried concept(s). Finally, the learning sequence is complemented with non-core related knowledge inferred using *Non CoreResourcesRule1&2* which are based on *HasPart* and its reciprocal relationship *IsPartOf*. Each of the above-mentioned rules checks whether the time needed to consume the recommended resource is within the learner's available time. It also checks for already covered concepts and consumed resources which are automatically eliminated from the learning sequence. Thus, making use of tacit knowledge retrieved from the learner profile.

Rule-Set-8 example of learning sequence construction
Ontology related facts

A1) before applying *SimilarResourceRule1:*

ConductedLearningActivity(Irene,A1) MakeQuery(A1, Logical Express)
HasKeyword(Logical Express, C_{28})
IsMappedTo(C_{28},LR_{28a})IsMappedTo(C_{28}, R_{28b})

B1) before applying *CoreResourceRule1*

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})

C1) before applying *NonCoreRelatedResourceRule2:*

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})

¬Consumed(Irene, LR_{5c})

Inferred facts

A2) after applying *SimilarResourceRule1*

SimilarLR(A1, LR_{28a})

SimilarLR(A1, LR_{28b})

B2) after applying *CoreResourceRule1*

CoreLR(A1,LR_{31b})

CoreLR(A1,LR_{32a})

C2) after applying *NonCoreRelatedResourceRule2*

NonCoreRelatedLR (A1,LR_{5a})

NonCoreRelatedLR (A1,LR_{5c})

To illustrate the reasoning mechanism adopted in the learner-adaptation process, we use a real life learning scenario based on the C++ Programming ontology shown in Figure 15. Let's assume that learner *Irene* wants to learn about "logic expressions" of the C++ programming language. This query has similar keywords with the ontology
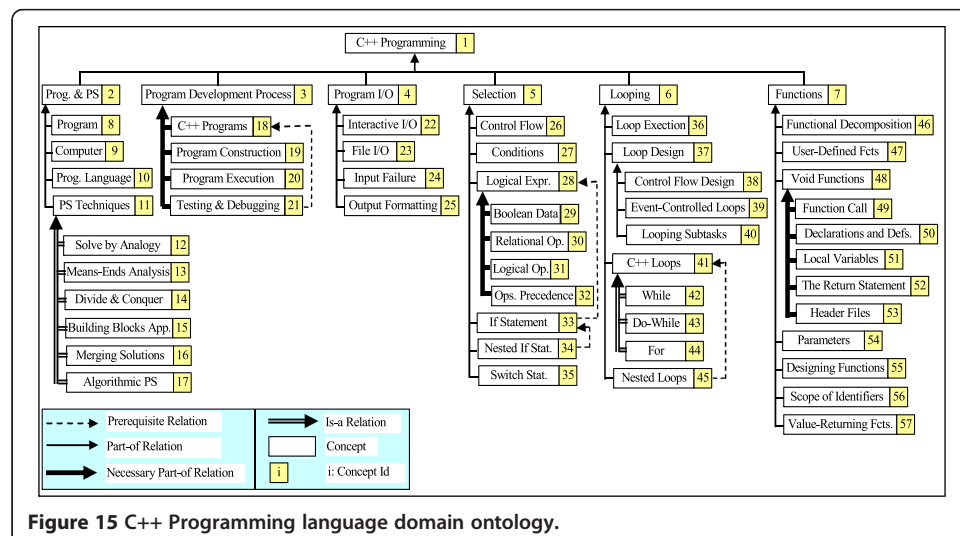
Figure 15 C++ Programming language domain ontology.

concept C28 which describes "Logical_Expression" as shown in Figure 15. The reasoning engine is invoked by firing *SimilarResourceRule1* which operates on facts (*A1*) and infers (*A2*) as shown in Rule-Set-8. The results of this inference consist of mapping learning resources *LR28a* and *LR28b* to concept *C28*. Facts (*A2*) are then added to the knowledge base. It should be noted here that concept *C28* does not have any similar or prerequisite concepts in the C++ ontology. Therefore, the application of *SimilarResourceRule2&3* and *PrerequisiteResourceRule* do not produce any useful results in this case. However, when applying *CoreResourceRule* on facts (*B1*), concepts *C29*(*Boolean_data*), *C30*(*Relational_Operators*), *C31*(*Logical_Operators*), and *C32* (*Operators_Precedence*) are inferred as core knowledge that need to be offered to the learner to fully understand the queried concept (i.e. *Logical_Expressions*). We also infer that concepts *C29* and *C30* have already been covered by Irene in previous studies, thus, the reasoner automatically eliminates them from the learning sequence. The system also infers that learning resources (*LR31a*, *LR31b*) and *LR32a* are mapped to concepts *C31* and *C32* respectively, and that learning resources *LR31b* and *LR32a* have not been consumed by *Irene*. These resources are therefore prescribed to *Irene* and facts (*B2*) are added to the knowledge base. Finally, *NonCoreResourceRule2* is applied on facts (*C1*) to infer (*C2*) which states that concept *C28*(*Logical_Expression*) is part of *C5*(*Selection*), and that *LR5a* and *LR5c* which correspond to concept *C5* have not been consumed by *Irene* so far, and therefore can be prescribed to her as non-core related knowledge. The learning sequence resulting from the application of the above mentioned rules produces the learning sequence shown in Figure 16 where the type of each learning resource (e.g. core, non-core, prerequisite, etc.) is clearly specified to the learner.

The learning path navigation algorithm, described below, is triggered once the learner starts interacting with the initially recommended learning sequence. The invocation of any learning resource leads to updating the list of consumed learning resources, and generating a new sub-learning path associated with the newly invoked concept. The new path is then added to the global learning path as shown in step 10 of the

Figure 16 Recommended learning sequence.
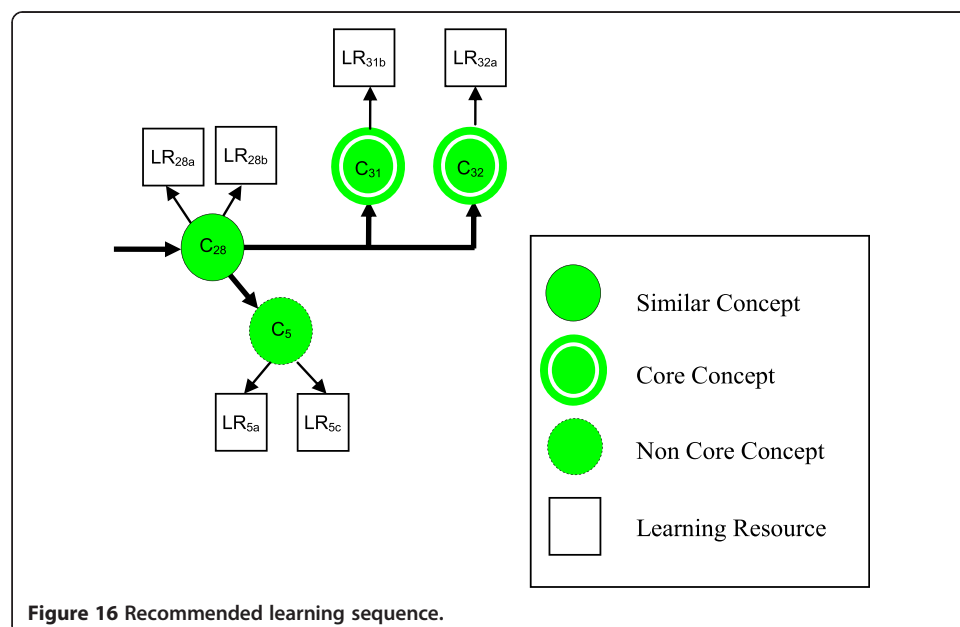
Learning_path_Navigation algorithm. The new sub-learning path is constructed by the *learning_path_generation* procedure, which first infers core concepts associated with the newly invoked concept and appends their respective resources. Then, for each core concept, it infers resources to fulfill their prerequisite knowledge. The learning path thus, grows dynamically as the learner invokes new concepts. However, when the learner makes a backward move, the corresponding sub-learning path is completely pruned from the global learning path. The learning session terminates when the learner either logs out of the system or consumes all prescribed learning resources.
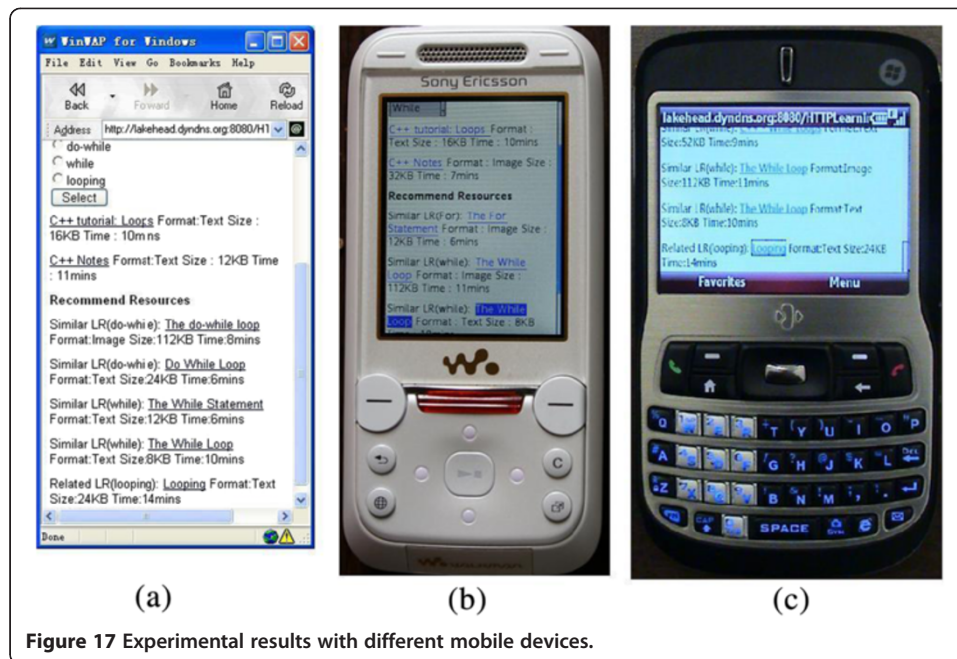
Experimental results

To illustrate the ontology reasoning mechanisms used in this study, we provide a number of scenarios that demonstrate the various system-centric and learner-centric adaptations. For system-centric adaptations, we devised an experiment where three mobile phones with different software and hardware capabilities were used to query concepts related to C++ programming language. Table 2 shows the software and hardware features of the used devices while Figure 17 shows the list of learning resources returned for each device after applying the various system centric rules described in section 4.1. The first device used in this experiment is an emulator for a Nokia phone with basic features and limited resources. The reasoning engine recognizes the limitations of the Nokia phone and the limited network bandwidth (8.0 kbps) sensed during the experiment, and returns a set of learning resources in text format, not exceeding 125Kbytes as shown in Figure 17.a. Figure 17.b shows the learning resources retrieved for the Sony Ericsson W830C which has more capabilities than the Nokia phone. These include resources of image media type in addition to text resources not exceeding 250Kbytes due to the limited sensed bandwidth. However, Figure 17.c shows the learning resources retrieved for the HTC S621 Smartphone which can support all types of media. Based on the sensed bandwidth and the predicted maximum resource size, the system was able to retrieve larger resources of various media types.

To demonstrate some of the learning centric adaptations used in this study, we show an experiment where Irene is working on a C++ programming assignment as part of her first programming course. Being a novice programmer, Irene is confused about the syntax of "C++ Loops". So, she used her mobile device "HTC S621 Smartphone" to query the system using "C++ Loops". Irene's available learning time is set to 15 minutes.

Table 2 Experimental Mobile Devices

| Feature	Basic Nokia Phone Emulator	Sony Ericsson W830C	HTC S621 Smartphone	Samsung S4 Smartphone
Operating system	Symbian	Sony Ericsson Java	Windows Mobile 6	Android 4.2.2
Available Memory	256.0kbytes	6.0Mbytes	32.0Mbytes	64GB
Connection Speed	32.0 kbps	48.0 kbps	120.0 kbps	150Mbps
Screen Resolution	128×96 pixels	320×240 pixels	320×240 pixels	1920×1080
Browser	WAP 2.0	WAP 2.0	Internet Browser	Internet Browser
Keyboard Type	Virtual	Virtual	Real	Virtual
Media Type	Text, Image	Text, Image, Video	Text, Image, Video	Text, Image, Video
Display Type	Monochrome	256 k Colors	65536 Colors	16 M Color Depth
Network Adaptor	GSM 1900	GPRS, EDGE	Wifi/GPRS/EDGE	Wifi/GPRS/EDGE/HSPA

Figure 17 Experimental results with different mobile devices.

Once logged in, the system tracks her previous login sessions, covered concepts, consumed learning resources, and previously conducted learning interactions. Then, the system proceeds with system-centric rules to adapt the search for learning resources, taking into account the limitations of the used device and surrounding environment. Irene's query is then used to reason with 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 ontology concepts *For*, *Do-While*, and *While* are inferred and classified as "similar knowledge" to C++ Loops, while concept *Looping* is inferred and classified as "related knowledge". As shown in Figure 15, the C++ ontology does not have prerequisite or core concepts associated with C++ Loops. Thus, the learning sequence suggested by the system for this scenario is C41 (C ++ Loops) → C44 (For) → C43 (Do-While) → C42 (While) → C5 (Looping) as shown in Figure 18.b. The system will then starts searching for learning resources associated with the inferred ontology concepts to build a personalized learning path. The learning path construction process uses the learner-centric rules described in section 5.2 to remove already covered concept and consumed learning resources. It also uses the learner's available time (i.e. 15 minutes) to suggest only those learning resources that can fit within the allotted time. Figure 18 shows Irene's interactions with the system for this learning scenario. The suggested learning resources for this case are a video about C++ loops from YouTube Pocket and a text tutorial about While loop as shown in Figure 18.

An additional experiment was conducted to show both system-centric and learner-centric adaptations using an ontology related to photography. The photography ontology edited using Protégé software is shown in Figure 19. In this scenario we assume that Irene would like to purchase a camera as a gift for her friend. On her way to the shopping mall she used our system to get some technical background about cameras in
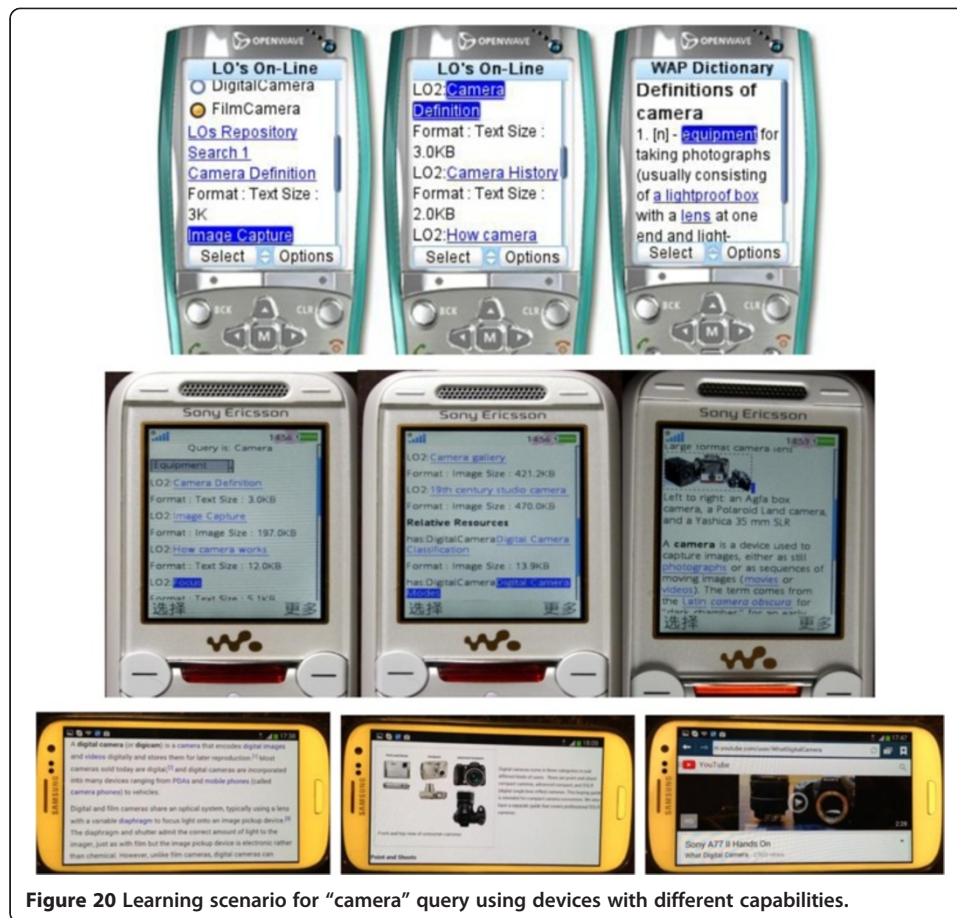
Figure 18 Learning scenario for C++ loops query.
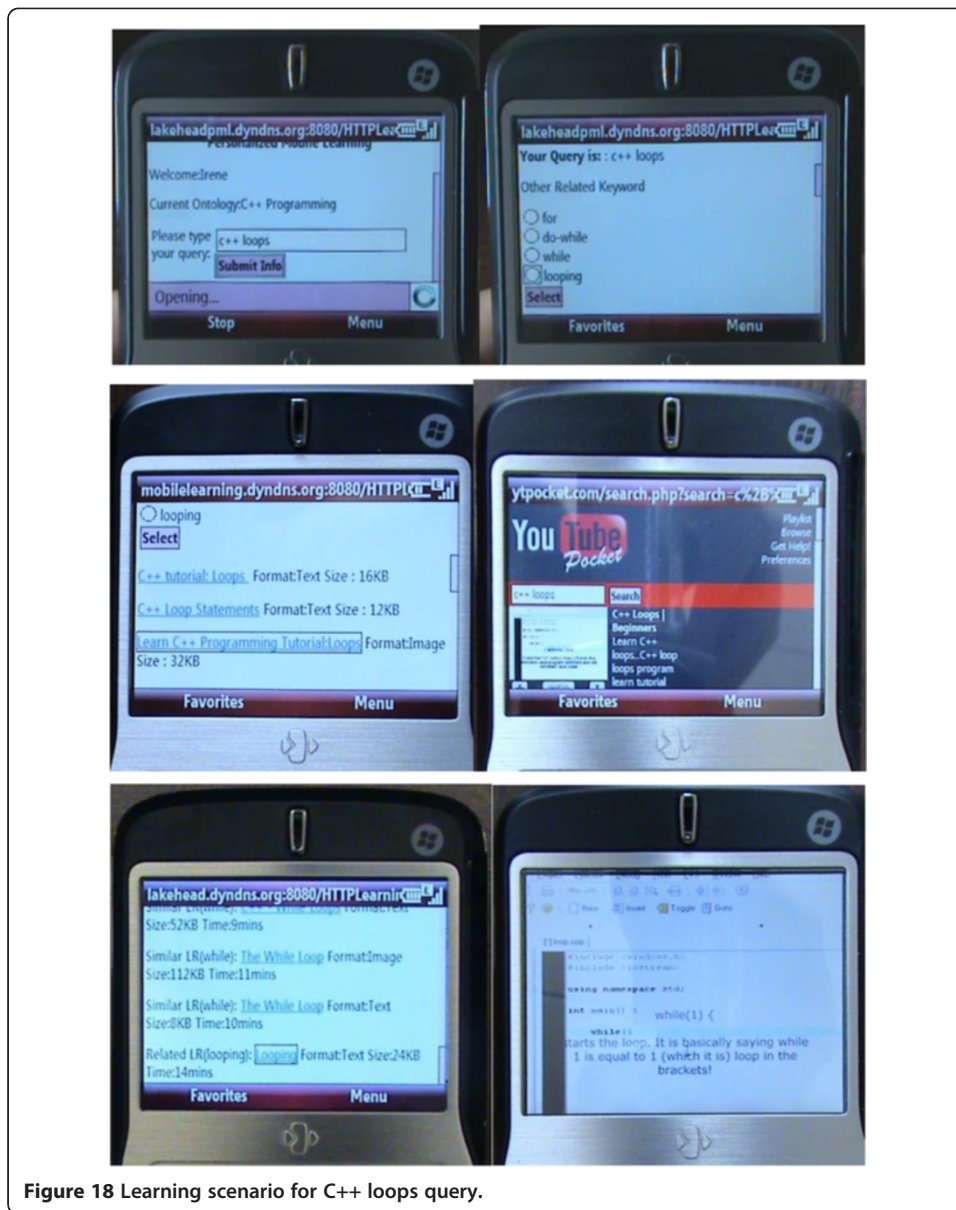
order to make the right purchase decision. She queries the system with the keyword "camera". This experiment is conducted using three different devices, the openwave simulator to emulate a Nokia phone with basic features, the Sony-Ericson W830C, and a Samsung S4 Smartphone. The capabilities and features of these devices vary from simple to sophisticated respectively as shown in Table 2. Irene's available learning time in this case is 20 minutes. For each experiment, first, system-centric adaptations rules are applied to select learning resources that can be played on the used device. Second, learner-centric adaptations rules are applied to infer concepts related to the user's query, taking into account the available learning time. The suggested learning concepts are inferred in the following order: similar knowledge, prerequisite knowledge, core knowledge, and then, related knowledge. In our experiment, the ontology concept *Digital Camera* and *Film Camera* are inferred as "similar knowledge" of concept *Camera.* The concept *Photography Equipment* is inferred as "related knowledge". Thus, the

Figure 19 Photography ontology.
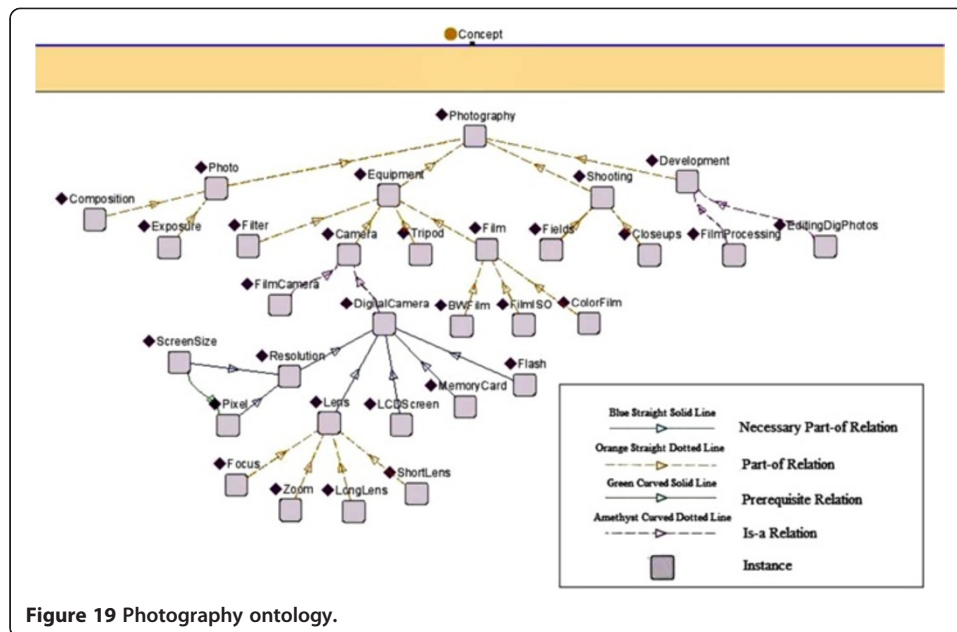
resulting learning sequence is (*Camera*) → (*DigitalCamera*) → (*FilmCamera*) → (*Equipment*). The learning resources associated with these concepts, as shown in Figure 20, are based on device capabilities and environment context as determined by the various adaptation rules. In this case, Irene is particularly interested in *Digital Cameras.* By choosing *Digital Camera,* further related concepts (*Resolution, Lens, LCD Screen, Memory Card, and Flash)* are inferred from the ontology and suggested for learning. Irene can then deepen her knowledge about any of these features before making a purchase decision. It can be seen in Figure 20 that resources suggested using the openwave emulator are mainly small text-based resources, while more image- and video-based resources are suggested for the Samsung S4 smartphone. The images for the Samsung are taken using the Samsung screenshot utility. Each time Irene interacts with the system, the learning resources associated with consumed concepts are removed and the learning path is updated accordingly as shown in Figure 20. Due to limited space we could not show all Irene's interactions, however, details of the above mentioned learning scenarios can be viewed in videos provided at http://unite.lakeheadu.ca/mlearning.

Conclusion

In this paper, a personalized mobile learning system that provides lightweight services adapted to both system-centric and learner-centric context is proposed. In particular, an attempt was made to solve two challenging problems related to context-aware mobile learning. These are heterogeneous context perception and integration at the semantic level, and context-aware content discovery and adaptation at run-time. To deal with context heterogeneity, an upper ontology space is used to define a higher level unique semantic view of the learning scenario. We showed that knowledge embedded in the upper ontology can homogeneously be used to enable a unified reasoning mechanism that operates on facts instantiated by the perceived heterogeneous context elements. The paper also addressed the problem of context that is perceived with ambiguity by using fuzzy logic in conjunction with semantic web based reasoning. The

Figure 20 Learning scenario for "camera" query using devices with different capabilities.

proposed system is characterised by the fact that the reasoning engine translates context changes as they occur into new content adaptation constraints in the operating environment, thus enabling context-aware personalized learning services. 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 learning content based on the learner's context and surrounding environment.

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 repositories, thus enhancing learning-content provision. In addition, we are currently working towards the provision of secured lightweight learning services. Trusted web-services are crucial for mobile learning applications such as those related to telemedicine or corporate learning.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The authors have significantly contributed in the research described in this manuscript. RB has made substantial contribution to the conception and design of the overall system developed in this research, while XZ has contributed in the design of the context acquisition and ontology, and has implemented most parts of the system. XZ has performed data acquisition and initial analysis of results, while RB has helped in the data analysis and significantly contributed in the interpretation of results. RB has drafted the manuscript and revise it critically for important intellectual content. XZ has helped drafting sections 3 and 4 of the manuscript. Both authors read and approved the final manuscript.

References
1. Teo CB, Gay RKL (2006) A knowledge-driven model to personalized e-Learning. ACM J Educ Resour Comput 6 (1):39–53
2. Davies J, Lytras M, Sheth A (2007) Semantic web based knowledge management. IEEE Internet Comput 10:14–16
3. Zouaq A, Nkambou R (2010) A Survey of Domain Ontology Engineering: Methods and Tools. In: Nkambou, Bourdeau, and Mizoguchi (ed) Advances in Intelligent Tutoring Systems. Springer-Verlag, Berlin Heidelberg. pp 103–119
4. Wu J (2010) An Ontology-Based Context Awareness Approach in Autonomous Mobile Learning. In: Proc. of Int. Conference on e-Business and e-Government ICEE2010, Guangzhou, China
5. Henze N, Dolog P, Nejdl W (2004) Reasoning and ontologies for personalized e-Learning in the semantic web. Educ Technol Soc 7(4):82–97
6. Benlamri R, Berri J, Atif Y (2006) A Framework for ontology-aware instructional design and planning. J E-Learn Knowl Soc 2(1):83–96
7. Huang T, Cheng S, Huang W (2009) A blog article recommendation generating mechanism using an SBACPSO algorithm. Expert Syst Appl 36(7):10388–10396
8. Yu Z, Nakamura Y (2007) Ontology-based semantic recommendation for context-aware e-Learning. Proc. Int. Conf. Ubuquitous and Intelligence Computing - UIC'2007. Lecture Notes in Computer Science 4611:898–907
9. Sudhana KM, Raj VC, Suresh RM (2013) An ontology-based framework for context-aware adaptive e-learning system. In: Proc. of the 3rd Int. Conference on Computer Communication and Informatics (ICCCI'2013), Chennai, India
10. Chia Y, Tsa FS, Tiong AW, Kanagasaba R (2011) Context-Aware Mobile Learning with a Semantic Service-Oriented Infrastructure. In: Proc. of 25th IEEE Int. Conference on Advanced Information Networking and Applications, Biopolis, Singapore
11. Hsu C, Hwang G, Chang C (2013) A personalized recommendation-based mobile learning approach to improving the reading performance of EFL students. Comput Educ 63(4):327–336
12. Weal MJ, Michaelides DT, Page K, de Roure DC, Monger E, Gobbi M (2012) Semantic annotation of ubiquitous learning environments. IEEE Trans Learn Technol 5(2):143–156
13. Dorn C, Dustdar S (2007) Sharing hierarchical context for mobile web services. Distr Parallel Databases 21(1):85–111
14. Bellavista P, Corradi A, Montanari R (2006) A mobile computing middleware for location and context aware Internet data services. ACM Trans Internet Technol 6(4):356–380
15. Wan N (2013) Conceptualizing m-learning literacy. Int J Mobile Blended Learn 5(1):1–20
16. McCalla G (2004) The ecological approach to the design of e-Learning environments: Purpose-based capture and use of information about learners. J Interact Media Educ 7:1–23
17. Mukherjee A, Saha D, Jha S (2003) Location management in mobile wireless networks. In: Handbook of Wireless Internet, Borko Furht Ed. Cat. No. 7006. CRC Press LLC, Florida, USA
18. Schilit B, Adams N, Want R (1994) Context-aware computing applications. In: Proc. of The IEEE Workshop on Mobile Computing Systems and Applications. IEEE Xplore, Santa Cruz, California, USA
19. Bu Y, Li J, Chen S, Tao X, Lv J (2005) An enhanced ontology based context model and fusion mechanism. Proc. Of the Int. Conf on embedded and ubiquitous computing. Lect Notes Comput Sci 3824:920–929
20. Dolog P, Henze N, Nejdl W, Sintek M (2004) The Personal Reader: Personalizing and Enriching Learning Resources Using Semantic Web Technologies. In: Proc. of 3rd Int. Conf. Adaptive Hypermedia and Adaptive Web-based Systems, Netherlands
21. Jeremic AZ, Jovanovic J, Gasevic D (2013) Personal learning environments on the social Semantic Web. J Semantic Web 4(1):23–51
22. Baldoni M, Baroglio C, Henze N (2005) Personalization for the Semantic Web. Proc. of Reasoning Web Conf. Lect Notes Comput Sci 3564:173–212
23. Kay J (2008) Life-Long Learning, Learner Models and Augmented Cognition. Proc. Int. Conf. Intelligent Tutoring Systems. Lect Notes Comp Sci 5091:3–5
24. Niu WT, Kay J (2008) Pervasive personalization of location information: personalized context ontology. Proc. Int. Conf. Adaptive hypermedia and adaptive Web-based systems. Lect Notes Comp Sci 5149:143–152
25. Siorpaes K, Simperl E (2010) Human intelligence in the process of semantic content creation. World Wide Web J 13(2):33–59
26. Dagger D, Wade VP, Conlan O (2004) Developing active learning experiences for adaptive personalized eLearning. Proc. Of Int. Conf. Adaptive hypermedia and adaptive Web-based systems. Lect Notes Comp Sci 3137:55–64
27. Conlan O, Wade VP (2004) Evaluation of APeLS - an adaptive eLearning service based on the multi-model, metadata-driven approach. Proc. Of Int. Conf. Adaptive hypermedia and adaptive Web-based systems. Lect Notes Comp Sci 3137:291–295
28. Yee KY, Ang WT, Tsai FS, Kanagasabai R (2009) OntoMobiLe: A Generic Ontology-Centric Service-Oriented Architecture for Mobile Learning. In: Proc. of 10th Int. Conference on Mobile Data Management, May 18–20, 2009, Taipei, Taiwan
29. Jimenez W, Vallina CPA, Gutierrez JMA, de Pablos PO, Gayo JEL (2013) Emergent Ontologies by Collaborative Tagging for Knowledge Management. In: de Pablos PO (ed) Advancing Information Management Through Semantic Web Concepts and Ontologies. IEEE Xplore, USA, pp 54–69
30. Zoua A, Ga D, Sevi C, Hatala M (2011) Towards open ontology learning and filtering. Inf Syst 36(7):1064–1081
31. Park J, Cho W, Rho S (2010) Evaluating ontology extraction tools using a comprehensive evaluation framework. Data Knowl Eng 69:1043–1061

32. Denaux R, Dolbear C, Hart G, Dimitrova V, Cohn AG (2011) Supporting domain experts to construct conceptual ontologies: a holistic approach. J Web Semantics 9(2):113–127
33. Dolog P, Nejdl W (2007) Semantic web technologies for the adaptive web: Methods and strategies of web personalization. In: The Adaptive Web. Lect Notes Comput Sci 4321:697–719
34. Jovanovic J, Gasevic D, Brooks C (2007) Using semantic web technologies to analyze learning content. IEEE Pervasive Comput 11(5):45–53
35. Krummenacher R, Kopecky J, Strang T (2005) Sharing context information in semantic spaces. Proc. Int. Workshop On the Move to Meaningful Internet Systems. Lect Notes Comp Sci 3762:229–232
36. Hage WRV, Katrenko S, Schreiber G (2005) A method to combine linguistic ontology-mapping techniques. Proc. 4th Int. Semantic Web Conference – ISWC'05. Lect Notes Comp Sci 3729:732–744
37. Mantyjarvi J, Seppanen T (2003) Adapting applications in handheld devices using fuzzy context information. Interact Comput 15(4):521–538
38. Pan JZ, Stoilos G (2005) f-SWRL: a fuzzy extension of SWRL. Journal on data semantics VI: special issue on emergent semantics. Lect Notes Comp Sci 3697:829–834
39. Tho QT, Hui SC, Fong ACM, Cao TH (2006) Automatic fuzzy ontology generation for semantic web. IEEE Trans Knowl Data Eng 18(6):842–856
40. Huang Y, Huang T, Wang K, Hwang W (2009) A Markov-based recommendation model for exploring the transfer of learning on the Web. Educ Technol Soc 12(2):144–162
41. Schmidt A (2008) Impact of Context Awareness on the Architecture of Learning Support Systems. In: Pahl C (ed) Architecture Solutions for e-Learning Systems. Idea Group Publishing, USA, pp 306–319
42. Hanaf HF, Samsudin K (2012) Mobile learning environment system (MLES): the case of android-based learning application on Undergraduates' learning". Int J Adv Comput Sci Appl 3(3):63–66
43. Hsu T, Ke H, Yang Y (2006) Knowledge-based mobile learning framework for museums. Electron Libr 24(5):635–648
44. Yarandi M, Jahankhani H, Tawi A (2012) An Ontology-based Adaptive Mobile Learning System Based on Learners' Abilities. In: Proc. of the Global Engineering Education Conference. Marrakech, Morocco, pp 17–20
45. Benlamri R, Zhang X, Berri J (2007) Proactive mobile learning on the semantic web. In: Int. Workshop on Ubiquitous Computing WUC'07, Portugal
46. Meawad FE, Stubbs G (2006) A framework for Interoperability with VLEs for Large Deployment of Mobile Learning. In: Proc. 4th IEEE Int. Workshop on Wireless, Mobile, and Ubiquitous Technology in Education WMUTE'06, IEEE Xplore, USA.
47. Kumar V, Merriman J, Thorne S (2002) What is the Open Knowledge Initiative? Eduworks Corporation white paper, September 20, 2002, http://mit.edu/oki
48. IMS Abstract Learning Framework white paper (2002) Revision July 31, 2002. http://www.imsglobal.org
49. Buttussi F, Chittaro L (2008) MOPET: A context-aware and user-adaptive wearable system for fitness training. Artif Intell Med 42(2):153–163
50. Ross TJ (2005) Fuzzy Logic with Engineering Applications, 2nd edn. John Willey and Sons Eds, New York.
51. Yang SJH (2008) Context aware ubiquitous learning environment for peer-to-peer collaborative learning. Educ Technol Soc 9(1):188–201
52. Benlamri R, Atif Y, Berri J (2008) An ontology-based approach for context-aware mobile learning. In: Advances in Ubiquitous Computing: Future Paradigms and Directions. Idea Group Publishing, USA, pp 23–44
53. CCPP Composite Capabilities Preference Profiles (2007). W3C (World Wide Web Consortium) standard, http://www.w3.org/Mobile/CCPP/
54. Cooper F (2007) Adapting to Available Network Bandwidth. Intel Software Network, Mobilizing Applications, 28 February https://software.intel.com/en-us/articles/mobilizing-applications-adapting-to-available-network-bandwidth-code-sample
55. IEEE LOM (2002) IEEE Learning Object Metadata Standard. http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html
56. Derballa V, Pousttchi K (2004) Extending knowledge management to mobile workplaces. In: Proc. Sixth International Conference on Electronic Commerce. Delft, The Netherlands
57. Rosati R (2006) The limits and possibilities of combining Description Logics and Datalog. In: Proc. 2nd Int. Conf. on Rules and Rule Markup Languages for the Semantic Web (RuleML'06), Georgia, USA, November 10–11, 2006
58. Rosati R (2008) On combining description logic ontologies and nonrecursive datalog rules. Proc. Of 2nd Int. Conf. On Web reasoning and rule systems. Lect Notes Comp Sci 5341:13–27
59. Mei J (2006) DatalogDL: datalog rules parameterized by description logics. Can Semantic Web Springer Series Semantic Web Beyond 2:171–187
60. Motik B, Sattler U, Studer R (2005) Query answering for OWL-DL with rules. J Web Semantics 3(1):41–60
61. Horrocks L, Boley PFH, Tabet S, Grosof B, Dean M (2004) SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, 21 May 2004, http://www.w3.org/Submission/SWRL/#2