ENABLING CONTEXT AWARE APPLICATIONS IN LEARNING ENVIRONMENTS

SOE LIN MYAT

B.Eng. (Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information, which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Soe Lin Myat

29 July 2014

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Bimlesh Wadhwa, for her guidance, support and understanding during my study. Without her, I would not have completed this thesis. Thank you, Prof!

I would also like to thank Dr. Atreyi Kankanhalli and Dr. David Rosenblum for their guidance.

Last but not least, I would like to thank my parents and Nang Mo for their supports and being there when I needed them during this period.

Table of Content

Summary

The increased prevalence of smart mobile devices presents myriad opportunities to utilize their unique capabilities in learning activities. Mobile devices not only allow learners to carry out learning activities anywhere and anytime, but also provide a platform for personalized and context-aware learning. Also, the use of mobile devices presents ways to increase the efficiency of classroom activities. For example, activities such as attendance taking and carrying out classroom polls can be automated through the use of mobile devises and appropriate context-aware applications. It will greatly reduce the time spent on these activities and give instructors more time to focus on other important learning activities.

This thesis reviews the state of the art in the context-aware learning, starting from the definition of context to key components in context-aware applications and proposes a general architecture for implementing context-aware learning applications. This thesis also discusses the use of context-aware applications in classrooms and proposes the Smart Classroom Framework (SCF) to provide the foundation for the rapid development and deployment of classroom context-aware applications.

Keywords: E-Learning, Context-Aware Learning, Computer-Assisted Instruction, Context-Aware Learning Support, Smart Classroom

List of Tables

List of Figures

Chapter 1 Introduction

1.1. Background

The wide usage of computers brings various applications to educational activities as teaching and learning support. For example, audio and visual aids are used to make learning more interactive and fun. PowerPoint slides are used to increase instructors' efficiency and improves the knowledge flow during classes. Various e-learning applications are developed to enable distant and personalized learning experiences. Integrated virtual learning environments are implemented to assist students and instructors in both administrative and learning activities. In a similar way, the increased prevalence of smart mobile devices presents myriad opportunities to provide further assistance in teaching and learning activities.

Mobile devices allow learners to carry out their learning activities anywhere and at any time. Also, mobile devices allow the learning environment to change as a learner moves from one location to another or enters different social situations. These devices can also capture the information of the context in which learning activities take place and detect changes in them. Such information includes learner's location, affective state, other devices in the environment, surrounding noise, light level, and social situations such as people and activities around the learner. This information provides learning applications with richer data for personalization such that new types of learning activities may be designed. For example, an English language learner can obtain learning content based on his current location (Chen and Li, 2010;

Hsieh, Chen, and Hong, 2007). If the learner is in a gym, the application may provide new words related to exercise or gym equipment to enhance his vocabulary learning. Context information can also be utilized in adapting the delivery of content. For example, learning materials can be provided to a learner in a suitable format for his or her device and Internet connection status (Gómez and Fabregat, 2010). If a fast Internet connection is detected, the application can provide a high quality video while low connection speed warrants an audio format. Such applications, which adapt to the context in which they are used, are called context-aware applications.

Moreover, instructors spend a large amount of precious classroom time on activities such as taking attendance, distributing learning resources, getting student feedbacks, carrying out and marking quizzes. With the use of smart mobile devises and context-aware applications, the time spent on these activities can be greatly reduced, giving instructors more time to focus on other learning activities. For example, a Bluetooth sensor in a classroom can automatically detect students in the class through students' mobile devices to take attendance without any input from instructors or students. Similarly, the context-aware application on a student's mobile device can detect the classroom that the student is currently in and download necessary and appropriate learning resources such as lecture notes. Other learning activities such as carrying out quizzes can be triggered using appropriate information such as time, location and students' information.

With the increasing use of mobile devices by students, context-aware mobile applications are now more relevant to educational activities than ever.

However, context-aware applications consist of various components and are complex in nature. The following are a few questions that need to be addressed when considering context-aware learning applications.

- How to model students, their knowledge level and their actions
- How to personalize the learning experience for students
- How to take advantage of contextual information, such as location, affective state, and social situations in learning applications
- How to design reusable and sharable learning resources

As such, it is worthwhile to review the state of the art in learning applications with regards to above questions.

Also, the widespread use of smart mobile devices among students may enable context-aware applications to improve classroom activities. It is worthwhile to explore classroom activities which may benefit from the use of context-aware applications and identify challenges of building classroom context-aware applications.

1.2. Thesis Objective and Scope

In the first part of the thesis, Chapter 2 and Chapter 3, we review the state of the art in the context-aware learning and propose an architecture for implementing context-aware learning applications. Our objective is to

(i) Provide a review of the evolution of e-learning applications and the literature on how context information can be used in learning applications

(ii) Consolidate our review and conceptualize Context-Aware Personalized Revision Aide (CAPRA).

In the second part of the thesis, Chapter 4 and 5, we review the use of Context-aware applications in classroom scenarios and develop a framework which can be used in developing classroom context-aware applications. Our objective is to

(iii) Provide an overview and challenges of developing Classroom Context-aware applications

(iv) Propose and develop the Smart Classroom Framework (SCF) which provides an extensible and scalable structure to enable efficient development and deployment of classroom context-aware applications.

1.3. Structure of the Thesis

The rest of this thesis is structured as follows.

- Chapter 2 reviews the state of the art in e-learning applications.
- Chapter 3 proposes the refined general architecture for context-aware learning applications and conceptualizes a contextual application to consolidate the reviews and ideas.
- Chapter 4 reviews the use of context-aware applications in classrooms and identify challenges in developing them.
- Chapter 5 proposes the Smart Classroom Framework (SCF) for the rapid development and deployment of classroom context-aware applications.
- Chapters 6 concludes this thesis by summarizing its contribution and discussing future directions

Chapter 2 Review on E-Learning Applications

2.1. Introduction

E-learning applications consist of various components and are complex in nature. Our contribution in this chapter is to provide a thorough survey of literature in e-learning concept and applications particularly, learner models, context models, learning design, personalization and context adaptation. Learner models and context models are chosen for the review as they are the building blocks of personalization and context adaptation. On the other hand, learner design is important for sharing learning resources among different learning applications. This chapter is structured as followed.

- Section 2.2 discusses the evolution of e-learning applications and various ways to classify them.
- Section 2.3 discusses the definition of context, starting from the very first introduction of the notion by Schilit, Adams and Want in 1994.
- Section 2.4 discusses learner modeling and how contextual information can be incorporated into the learner model.
- Section 2.5 discusses Learning Design which enables reusable and sharable learning resources
- Section 2.6 discusses personalization and context adaptation in learning applications
- Section 2.7 summarizes the chapter

2.2. Classification of E-Learning Applications

An e-learning application is defined as the delivery of educational activities or content to learners by electronic means. Researchers have categorized e-learning in various ways based on different criteria of e-learning. For example, based on the equipment or devices involved, e-learning has been classified into multimedia learning, computer-based learning, ubiquitous learning and mobile learning. A large number of e-learning applications such as Technological Enhanced Learning System (Goodyear and Retalis, 2010; Heeter, 1999; Mor and Winters, 2007), Intelligent Tutoring System (Brusilovsky, Schwarz, and Weber, 1996; Clancey, 1982; Graesser, Chipman, Haynes, and Olney, 2005), Adaptive Educational System (Kelly and Tangney, 2006; Shute and Zapata-Rivera, 2012; Triantafillou, Pomportsis, and Georgiadou, 2002), Web-based Training System (Barron, 1998; Horton, 2000), and Recommendation System for E-Learning (Sanjuan-Martinez, G-Bustelo, Crespo, and Franco, 2009; Shishehchi, Banihashem, and Zin, 2010) have been proposed over the past few decades.

To avoid confusion, we consistently use the categorization of e-learning systems by Das, Bhaskar, Chithralekha, and Sivasathya (2010) in this thesis. This categorization is chosen as it covers all types of e-learning systems and also considers context-awareness, which is of interest in this thesis. According to their work, e-learning falls into three categories - (i) Traditional E-Learning, (ii) Personalized E-Learning, and (iii) Context-aware E-Learning, as shown in Table 1.

Table 1: Classification of Electronic Learning

Classification of Electronic Learning	**Characteristic**	**Example Applications**
Traditional E-Learning	No personalization. All students receive the same learning materials	CD-ROM and DVDs, Cassette tapes, Screen casts of lectures
Personalized E-Learning	Learning materials are recommended or personalized based on Learner Model which includes his or her knowledge	Personalized Intelligent Mobile Learning System (Chen, Hsu, Li, & Peng, 2006), Learning Intelligent Advisor (Capuano, et al., 2009)
Context-aware E-learning	Learner's current situation and surrounding is considered in addition to Learner Model in recommending and adapting learning materials	Context-Aware Mobile Learning English System (Viet Anh, et al., 2010), Personalized Context-Aware Ubiquitous Learning System (Chen & Li, 2010)

Traditional E-Learning provides all learners with identical material. It does not consider individual learner's needs, knowledge level, or goals and is the most rudimentary form of electronic learning. Traditional E-Learning is not limited to online education and can involve other electronic equipment such as CD-ROMs.

Personalized E-Learning refers to an educational model that is customized for individual learner's interests and needs. It personalizes learning activities based on a “Learner Model” which includes learner’s interests, knowledge, background, goals and individual traits (Brusilovsky and Millán, 2007) (refers to section 2.4.1). However, it does not take into account of a learner's current situation.

Context-aware E-learning selects or filters learning resources to provide relevant or suitable information according to a learner's context. For example, the Context-Aware Mobile Learning English System (CAMLES) proposed by Viet Anh, Cong, and Dam (2010) provides adaptive context for different learners based on location, manner, time as well as learner's knowledge.

This thesis focuses on context-aware applications in education. In the next section, we discuss the definition of context, starting from the very first introduction of the notion by Schilit, Adams and Want in 1994.

2.3. What is Context?

Schilit, Adams, and Want (1994) first introduced the term “context-aware”, in which various mobile, stationary, and embedded computers are employed by users throughout the day and found that the execution environment (context) changes constantly. The challenge is to exploit the context with a new type of applications i.e., context-aware applications which recognize their current context and adapt to changes to the context. The paper defines user’s proximate environment in terms of location of the user, other people with the user and the resources nearby. The execution environment (context) however is composed of three components - (i) computing context, (ii) user context and (iii) physical context. Computing context deals with technical aspects such as communication cost, network connectivity, communication bandwidth, and nearby resources such as displays, printers and workstations. User context refers to information such as the user’s location, profile and the social situation such as the people and activities nearby. Physical context deals with aspects that represent the real world such as noise levels, lighting, temperature and traffic conditions.

Abowd, et al. (1999) consider the definition of context in Schilit et al. (1994) to be too narrow. They argue that situations differ greatly from one another and that which context is important in a particular situation cannot be generalized. As a result, they define context broadly as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” (p304). They believe it should be left to application developers to decide which information constitutes context for a given application scenario. They also argue that the general assumption of context as implicit information is troublesome and that their definition allows context to be indicated by the users either implicitly or explicitly.

Alternatively, context can be defined as "the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user" (p3) (Chen and Kotz , 2000). Time is also added as another dimension of context. Time context refers to time of the day, week, month, season of the year and whether a particular event is sporadic or periodic.

Context has been classified as primary or secondary by Abowd et al. (1999). Secondary contexts are defined as contexts that can be derived from primary context. The information of nearby people is a secondary context as it can be derived from users' locations, which is a primary context. Abowd et al. (1999) classify location, time, identity and activity as primary contexts and all other types of context as secondary contexts.

Context can also be classified as either active or passive based on how it is used in an application (Chen and Kotz, 2000). Active context influences the behavior of an application while passive context, although captured, does not impact the application's behavior. For example, in a video streaming application, the Internet speed is considered as an active context if the quality of the video delivered is based on it. On the other hand, in a social application which lists the friends who are nearby, the location is considered as a passive context as it does not affect the behavior of the application directly.

Context models are designed to contain contextual information in applications. In the educational domain, there is no commonly accepted standard on what constitutes a learner’s context model. We discuss more on it in the next section.

2.4. Learner Modeling

2.4.1. Learner Model

The Learner Model is also variously referred to as “Student Model” (Beck, Stern, and Haugsjaa, 1996; De Arriaga, Gingell, De Arriaga, Arriaga, and Arriaga, 2008; Schiaffino, et al., 2008), “User's Model” (Falquet and Ziswiler, 2005) or “User's Profile” (Kritikou, et al., 2008; Rumetshofer and Wob, 2003). To avoid confusion, we consistently use the term "Learner Model" in this thesis. The Learner Model is used in the adaptation and personalization process of learning applications.

We discuss here “Learner Model” (Schiaffino, Amandi, Gasparini, and Pimenta, 2008; Zapata-Rivera and Greer, 2001) which is commonly used in Personalized E-Learning applications.

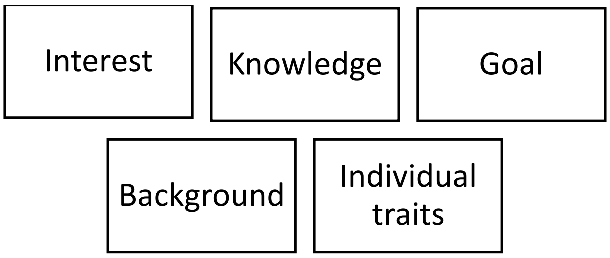


Figure 1: Five popular features of a Learner Model

The five most popular and useful features of a Learner Model are learners' interests, knowledge, goals, background, and individual traits (Brusilovsky and Millán, 2007). Learner’s knowledge represents learner’s expertise level of a learning domain, while the learner's interest is used to increase their motivation to learn. The learner's goals or tasks represent the immediate purpose of the learner's work. The learner's background is related to the learner's past experience outside the main area of the application such as learner's profession, job responsibilities, and work experience. Individual traits define the learner as an individual and include personality traits, cognitive styles, cognitive factors, and learning styles.

Among the five features, learner's knowledge is considered the most important feature (Brusilovsky, 1994) of the Learner Model. It can change dynamically within a learning session or between sessions. A learning application therefore would need to update the Learner Model when the learner’s knowledge changes. The simplest form of a learner’s knowledge model is the scalar model (Antal and Koncz, 2011). It estimates the learner’s knowledge level to a single qualitative or quantitative value. For example, a learner's knowledge on a topic, say, "Design Thinking" could be rated on a scale of 1 to 3 quantitatively or as “bad”, “average” and “good” qualitatively. In both cases, the learner's knowledge regarding the lesson is specified as a single scalar value. Although the scalar model is simple and easy to implement, it has low precision of a learner’s knowledge.

Alternative to the scalar model is the structural model. One popular structural model is the overlay model, which divides a lesson into independent elements and stores information about each element (Antal and Koncz, 2011). For example, a lesson on the topic "Design Thinking" could be divided into "Inquire", "Integrate", "Invent", and "Innovate", and a scalar model can be applied on each element. Another simple model is the stereotype student model which classifies students into several typical stereotypes (Sampson, Karagiannidis, and Kinshuk, 2010). For example, a student can be a Beginner, Intermediate, or an Expert student.

2.4.2. Context Model

Learner modeling in context-aware applications adds a learner's contextual information to the Learner Model. In the literature, various terms exist to define this combination, the common ones being context model or context. To avoid confusion, we introduce the term “Learner's Context Model” and use it to describe the learner’s model in context-aware learning. Currently, there is no commonly accepted standard on what constitutes a Learner's Context Model. In this section, we discuss recent attempts by researchers towards standardization.

Siadaty, et al. (2008) proposed to broadly divide context in m-learning into two parts: (i) Learning Context and (ii) Mobile Context. Learning Context refers to aspects related to the learning design. Mobile Context deals with the mobile environment with which learners interact to complete learning activities.

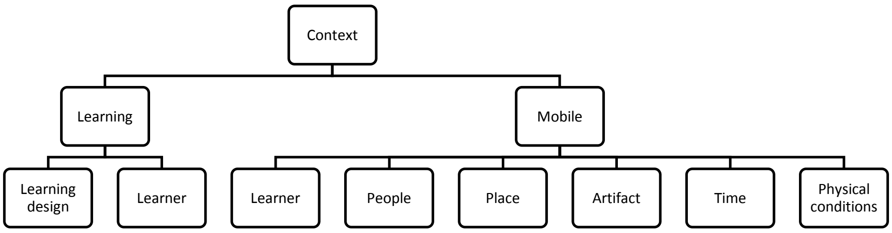


Figure 2: Context in m-learning

Studies have gone further to describe the Learning Context and Mobile Context. Zervas, Ardila, Fabregat, and Sampson (2011) proposed two dimensions for the Learning Context and six dimensions for the Mobile Context. The dimensions for the Learning Context are (i) Learning design, and (ii) Learner. The Learning design dimension deals with (a) learning objectives, (b) learning activities, (c) pedagogical models, (d) participating roles, (e) resources and (f) tools. The Learner dimension deals with (a) role, (b) competence profile, and (c) semi-permanent personal characteristics. The dimensions for Mobile Context are (i) Learner, (ii) People, (iii) Place, (iv) Artifact, (v) Time, and (vi) Physical conditions. Learner dimension in the Mobile Context deals with temporal personal information such as mood and temporary interests. People dimension refers to the relationship, role, constraints and contributions. Place dimension is concerned about zones, location, cultural background, interactive space and the learning setting. Artifacts are technological elements such as the physical and digital properties, and non-technical elements. The time dimension includes task scheduled, duration and when action happens. Physical conditions deal with real world aspects such as illumination, noise level, and weather conditions.

Alternatively, Tankeleviciene and Damasevicius (2009) proposed to separate Learning Context into seven different levels. They are (i) technological, (ii) pedagogical, (iii) e-learning methodology, (iv) organizational, (v) psychological, (vi) subject domain, and (vii) course. The technological level composes the hardware, networking, software, and user interface aspects. The pedagogical level is concerned with learning theory and instructional strategy. E-learning methodology deals with delivery models, e-learning form, and interactivity level. The organizational level represents study types such as formal and informal learning. The psychological level deals with psychological aspects of a learner such as motivation and preferred senses. The subject domain level defines structured-ness and didactics. The course level deals with the aims of learning and previous experience. This model focuses on the pedagogical and psychological side of e-learning processes and does not cover the learner's attributes in detail.

Another approach is proposed by Das et al. (2010), who seeks to define a Context Model to establish a learner’s context completely. They develop the model by consolidating various context parameters from existing literature and organize them into three context categories: (i) personal context, (ii) abstraction context, and (iii) situation context. Personal context includes personal type and information, and the knowledge level of the learner. Abstraction context includes learner intention, preference, and the style of learning. Situation context includes learner network, device, situation, and quality of learning service (QoLS). However, this model does not consider psychological aspects and the learner’s cognitive level and also does not support scenarios where multiple learners engage in the same learning activity unlike the previous two models that we have discussed above.

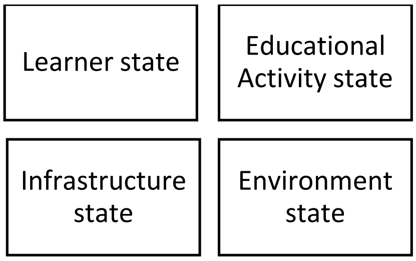


Figure 3: Four states of context by Economides

Alternatively, Economides (2009) defines learner’s context to consist of (i) the Learner state, (ii) the Educational Activity state, (iii) the Infrastructure state, and (iv) the Environment state. At a given moment, different infrastructures in multiple environments are used by multiple learners to perform educational activities. Thus, the complete context description would include all the interconnections. The Learner's state consists of 25 dimensions about the learner such as demographic, preferences, previous achievements and rewords, while the Educational Activity's state consists of 22 dimensions such as subject, keywords and educational level. The Infrastructure's state is divided further into three sub-groups: (i) Device, (ii) Network, and (iii) Other Hardware and Software Resources. The Environment's state consists of five dimensions such as Terrain and Neighbors. Economides identifies some of the dimensions and variables as fixed while others as adaptable. For example, Learner's Favorites and Interests are fixed and declared by the Learner. On the other hand, dimensions such as Participants and Teams, Presentation and Media, Sequencing and Feedback are adaptable.

When comparing these different models, we find that the Economides' model is more comprehensive than the other models reviewed here. Intuitively, one may infer that with a more detailed representation of context, comprehensive context-aware applications can be developed. Nevertheless, it is important to realize that the complexity of an application would grow with the extent of context representation. Technical challenges may also prevent the accurate acquisition of certain contextual information such as location. Typically, an application would require only a subset of the context dimensions proposed by Economides (2009).

2.5. Learning Design

A key factor in designing learning applications is the reusability and sharing of learning resources and activities. For this purpose, it is important to use a standardized language design to describe learning resources and activities such that they can be reused and shared across different learning applications. Learning design (LD) is formally defined as "the systematic process of translating general principles of learning and instruction into plans for instructional materials and learning" (Koper, 2005a). There are two widely used Learning Design standards: (i) Sharable Content Object Reference Model (SCORM) and (ii) IMS LD.

Sharable Context Object Reference Model (SCORM) is a collection of standards. It aims to describe content objects, data models, and protocols which can be shared between different systems using the same models (ADL, 2004). It was developed by the Office of the United States Secretary of Defense through the Advanced Distributed Learning (ADL) Initiative to solve a number of problems in web-based e-Learning such as incompatibility of learning resource formats between different systems. The initial purpose was to create a web-based mechanism which is reusable and sharable. It aims to reduce the time and cost for content creation and encourage sharing of content among different systems. However, the weakness of SCORM is that it can only deal with self-paced learning materials and does not provide specifications for multi-role collaborative and interactive learning required by the web 2.0 era. IMS LD deriving from the Instructional Management Systems (IMS) project addresses this inadequacy (Qu and He, 2009).

IMS LD was approved in 2003 by the IMS Global Consortium. It is a learning activity framework which includes various standards such as IMS Content Package, IMS Simple Sequencing, IMS Question and Test Interoperability, IMS Meta-Data, IMS Learner Information Package, IMS Reusable Definition of Competency or Educational Objective and IMS Enterprise. The key advantage of IMS LD is that it only requires one set of tools for learning applications for describing different pedagogies. IMS LD has three levels of implementation and compliance. Level A contains the majority of IMS LD constructs such as activities, plays, and roles. Level A is extended by adding properties and conditions in Level B while notifications are added to Level C (IMS, 2003).

Compared to SCORM, IMS LD can be used to describe a learning environment with either one or multiple learners and is flexible about learner grouping. It focuses on learning activity structure rather than the learning content. Further, learner interaction typically occurs in the form of discussion forums and chat rooms, or is supported by simulation and self-test. It also supports the personalization of the learning route (Qu and He, 2009). Numerous learning applications have used IMS LD (De Jong, Specht, and Koper, 2007; Gómez and Fabregat, 2010; Van Rosmalen, et al., 2005). Zervas, et al. (2011) describe design requirements for IMS LD authoring and player tools that incorporate the content adaption mechanism.

2.6. Personalization and Context Adaptation

2.6.1. Personalization of E-Learning Applications

In Personalized E-Learning applications, personalization happens through sequencing of learning activities or learning content recommendation to suit each individual learner. Personalization is based on the Learner Model which includes a learner's characteristics such as goals, knowledge level, background, interest, preferences, stereotypes, cognitive preferences, and learning styles (Ruiz, et al., 2008). We present an overview of three different examples of personalization in existing applications below.

Yu, et al. (2007) proposed an Ontology-based semantic content recommendation. It has three ontologies: Learner, Learning Content, and Domain Ontology. Learner Ontology represents context about a learner such as mastered content, available learning time, location and learning goal, interests and style. Learning Content Ontology defines properties of contents and the relationships between them. For example, the relationship hasPrerequisite describes content dependency information. The Domain Ontology combines existing ontologies such as mathematics, chemistry and computer science. The topics are classified in a hierarchical structure e.g., based on the ACM taxonomy for computer science. The learning content recommendation consists of four parts. First, in the Semantic Relevance Calculation step, the semantic similarity between a learner and learning materials is computed. The materials are then listed in descending order of similarity. Second, in Recommendation Refining step, learners can refine the results until they are satisfied with the options. Third, when one item is chosen by the learner from the list, a studying path is generated including prerequisite and target learning materials in the Learning Path Generation step. Finally, the Recommendation Augmentation adds the related materials to the main course. Information on the learner, learning materials and domain is used in each step of recommendation.

Capuano et al. (2009) proposed Learning Intelligent Advisor (LIA), a tutoring engine, based on their previous work (Capuano, et al., 2002). LIA uses four models - domain model, learner model, learning activity model, and unit of learning to provide personalized learning experiences in relation to learning objectives, preferences and current knowledge. Domain model describes learning objects in terms of a set of concepts and relations between them. Three possible relations between concepts are BT (belongs to), IRB (is required by), and SO (suggested order). The domain model is represented by the concepts graph G(C, BT, IRB, SO). The learner model consists of a cognitive state and a set of learning preferences. The unit of learning describes a set of learning activities that a learner needs to undergo to learn a particular target concept (TC). LIA generates the learning activities sequence by finding a learning path, starting from TC and a domain model, considering the concepts graph G(C, BT, IRB, SO).

Our last example shows the use of repertory grid in the personalization process. Hsu, et al. (2013) suggested a personalized recommendation-based mobile learning application which provides reading material for English as a Foreign Language (EFL) learners based on their preferences and knowledge levels. The recommendation mechanism depends on the repertory grid as a knowledge acquisition method. A repertory grid is a matrix, which uses a set of constructs to describe the similarities or differences between elements. In their experiment, Hsu, et al. (2013) used ninety five elements in the repertory grid. Another repertory grid was used to describe the learners' preferences based on a questionnaire. A similarity formula was developed to recommend most suitable articles to learners.

These are just a few examples of how personalized learning can be provided. In fact, many different algorithms have been proposed and used for the purpose of personalization (e.g. Yu, Nakamura, Jang, Kajita, and Mase, 2007; Shishehchi, et al., 2010; Shishehchi, Banihashem, Zin, Noah, and Malaysia, 2012).

2.6.2. Context Adaptation

Context Adaption can be related to learning activities or educational resources (Sampson, et al., 2012). In relation to learning activities, it affects sequencing and recommendation of learning activities and content, similar to Personalization described in the section 2.6.1. For example, Chen and Li (2010) present an English vocabulary learning system, called (PCULS), which provides adaptive English vocabulary learning, depending on learner's location, learning time, leisure time, and personal vocabulary abilities. In this case, the provided learning content is directly affected by the learner's contextual information. Context Adaptation is also related to educational resources such as resources types. When designing Context-aware mobile learning applications, one would need to consider technical capability of different devices such as screen-size, processing power, and memory. For example, whether to use an image or a video and what size to use would depend on the technical capabilities of the mobile device used.

Gómez and Fabregat (2010) address this issue and propose a design structure that provides contents automatically adapted to a learner's mobile device based on device limitations. This is achieved by carrying out the adaptation process at both design-time and run-time. Their structure uses IMS LD as the learning design. Their adaptation process at design-time involves 3 phases: (i) Unit of Learning (UoL) edition phase, (ii) content preparation and evaluation phase, and (iii) adapted content creation adjustment phase. During the UoL edition phase, the author can edit and build a UoL for the course curriculum following the IMS LD guidelines. He or she can specify available learning resources based on the conditions of the learner's situation using IMS LD conditional structures. When a UoL is uploaded, the application moves to the content preparation and evaluation phase. In this phase, the application generates transcoding requests to create different sets of resources based on predefined adaptation rules which specify the formats of the resources that are suitable to the profile of the devices that are supported by default. The adapted content creation and adjustment phase then creates the requested resources. As the outcome of this phase, new UoL structures (one per each default delivery device profile) are built and new adapted versions of the resources are adjusted to them.

The adaption process at run-time includes 3 phases as well: (i) detection phase, (ii) validation and content preparation phase and (iii) adapted content adjustment and delivery phase. In the detection phase, the context information related to the device, place, time and physical environment are detected. In the validation and content preparation phase, it is determined whether the learner’s mobile device is capable of accepting the resources transcoded at design-time. If the device is compatible, previously adapted resources at design-time are immediately delivered. If not, the incompatibilities of the device are determined to generate more detailed transcoding requests. In the third phase, the adapted UoL and corresponding resources are then created and delivered to the learner's device.

Based on the work by Gómez and Fabregat (2010), Gomez, Zervas, Sampson, and Fabregat (2012) discuss three possible types of adaptations based on contextual information. These include (i) Learning Activity Adaptation, (ii) Learning Content Adaptation and (iii) Learning Tools and Services Adaptation. The adaptations are done using polymorphic presentation and filtering mechanisms. The polymorphic presentation mechanism handles educational resources transformation as discussed in the previous paragraphs. The filtering mechanism uses IMS LD Level B properties and conditions to describe a pre-defined decision tree to handle contextual elements.

2.7. Summary

In this chapter, we have reviewed the state of the art in e-learning applications. We have reviewed various ways to classify e-learning applications and the definition of context. We have also reviewed the literature on learner modeling, Learning Design and various ways to carry out personalization and context adaptation. Table 2 briefly summarizes our review.

Table 2: Summary of the review on e-learning applications

Topic	**Classification**	**Explanation**
Classification of e-learning applications	Traditional	All students receive the same learning materials.
	Personalized	Learning materials are personalized based on student’s knowledge and interest.
	Context-aware	Learner’s current situation is considered in recommending learning materials.
Learner modeling	Learner Model	Learner model includes interests, knowledge, goals, background, and individual traits.
	Context Model	Context model concerns with a learner's contextual information.
Learning Design	SCORM	Both IMS LD and SCORM are collections of standards developed with the aim to share learning resources across different system
	IMS LD	
Personalization and Context Adaptation	Personalization	Personalization happens through sequencing or recommendation of learning activities or content to suit each learner.
	Context Adaptation	Context information is used in recommending learning materials.

Chapter 3 Architecture for Context-Aware Learning applications

3.1. Introduction

In the previous chapter, we have reviewed the state of the art in e-learning applications and discussed their various key components. In this chapter, our main contribution is to extend and refine an existing architecture for context-aware mobile learning applications and to conceptualize a context-aware application, CAPRA, to consolidate our literature review so far. This chapter is structured as follow.

- Section 3.2 proposes the refined general architecture for context-aware mobile learning applications
- Section 3.3 discusses a conceptual IMS LD Level-C compliant context-aware learning application, Context-aware Personalized Revision Aide, to consolidate our review and ideas.
- Section 3.4 summarizes the chapter.

3.2. Architecture for Context-aware Mobile Learning Applications

In this section, we propose an extended and refined architecture for context-aware mobile learning applications (see Figure 4). The original architecture by Sudhana, Raj, and Suresh (2013) is designed for Ontology-based applications. However, we generalized the architecture for any type of context-aware mobile learning applications. We have also added new components, Notification and Learning Objects/Activities Editors to complete the architecture.

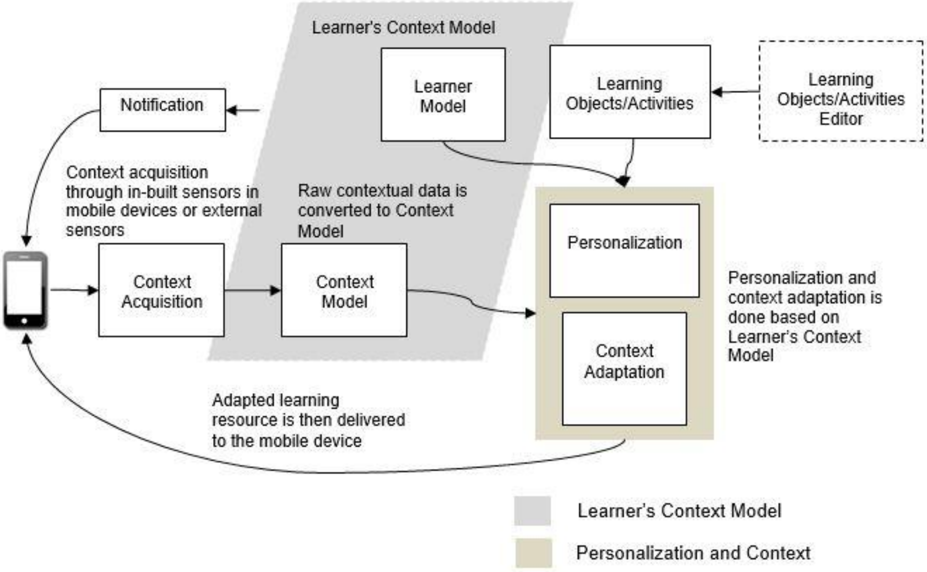

Learner's Context Model
Learner Model
Learning Objects/Activities
Learning Objects/Activities Editor
Notification
Context acquisition through in-built sensors in mobile devices or external sensors
Raw contextual data is converted to Context Model
Personalization
Context Acquisition
Context Model
Context Adaptation
Personalization and context adaptation is done based on Learner's Context Model
Adapted learning resource is then delivered to the mobile device
Learner's Context Model
Personalization and Context

Figure 4: Proposed Architecture for context-aware mobile learning applications

The architecture involves 8 main components - (i) Learner Model, (ii) Context Acquisition, (iii) Context Model, (iv) Learning Objects/Activities, (v)

Personalization, (vi) Context Adaptation, (vii) Learning Objects/Activities Editor, and (viii) Notification as described below.

Learner Model stores a learner's characteristics such as goals, knowledge level, background, interest, preferences, stereotypes, cognitive preference, and learning style (Ruiz, Diaz, Soler, and Perez, 2008). The Learner Model can be captured explicitly from the learner or by analyzing the learner's interactions with the application (Verbert, et al., 2012). Another potential way to capture information for the Learner Model is from third-party sources such as the learner's online social profiles.

Context Acquisition can be done through built-in sensors of mobile devices such as the GPS, accelerometer, proximity sensor, ambient light sensor, and gyroscope. Context can also be acquired using external sensors or embedded objects such as RFID tags in the learning environment (Chin and Chen, 2013; Hwang, Tsai, and Yang, 2008; Hwang, Yang, Tsai, and Yang, 2009). This type of learning which involves additional external sensors and embedded objects to allow students to totally immerse in the learning environment is called ubiquitous learning. In this paper, our focus is on context-aware learning rather than ubiquitous learning.

Context Modeling converts raw contextual data into the learner's Context Model. Researchers have proposed various middleware platforms to handle context acquisition and modeling (Carlson and Schrader, 2012; Thüs, et al., 2012; Zhu, et al., 2011). In this paper, we will use the term "Learner's Context Model" to describe the combination of the Learner Model and Context Model as shown in Figure 4.

Learning Objects / Activities are usually represented in a learning design standard such as SCORM or IMS-LD. These standards are important for sharing and reusing educational resources across different learning applications. We have discussed SCORM and IMS-LD in section 2.5. A Learning Objects/Activities Editor allows instructors to create or edit learning objects and activities. Custom-built editor or open-source editors can provide instructors with a visual authoring environment for creating Learning Objects/Activities and support language design. IMS-LD Reload (Milligan, Beauvoir, and Sharples, 2005), OpenGLM (Derntl, Neumann, and Oberhuemer, 2011), LAMS (Dalziel, 2003) are a few examples of open source editors.

Personalization for learning activities uses the information stored in the Learner Model. Personalization can take place in the form of sequencing of learning activities (Capuano, et al., 2009; Capuano, Gaeta, Micarelli, and Sangineto, 2002; Capuano, Gaeta, Salerno, and Mangione, 2011) and content recommendation (Shishehchi, et al., 2010; Shishehchi, Banihashem, Zin, Noah, and Malaysia, 2012; Yu, Nakamura, Jang, Kajita, and Mase, 2007). Learning activities consist of units of learning, which are the smallest chunks in an activity. These units can be sequenced differently based on the learner's knowledge to provide a personalized learning experience. Content recommendation provides personalized help and exercises in addition to the learning materials.

Context Adaptation uses the Context Model to filter or adapt learning resources to provide relevant or suitable information according to a learner's context (Das, et al., 2010). Adaptation can be related to educational resources or learning activities (Sampson, Isaias, Ifenthaler, and Spector, 2012). For example, educational resources can be adapted according to the surrounding situations such as lighting, noise level and internet connection strength. The adapted learning resource is then delivered to the mobile device of the learner.

Notification is a mechanism to notify a user if a certain event occurs. For example, in a social learning application, the server sends a notification to a user through the mobile application if a peer is nearby.

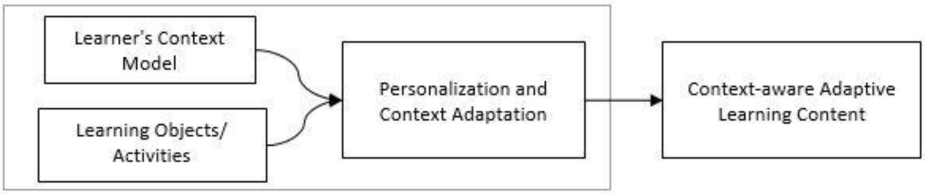

Learner's Context Model
Learning Objects/ Activities
Personalization and Context Adaptation
Context-aware Adaptive Learning Content

Figure 5: Context adaptation in context-aware learning applications

We can view the Learner's Context Model and Learning Objects/Activities as inputs into the Personalization and Context Adaptation process, and the Context-aware Adaptive Learning Content as the output as shown in Figure 5. Learning Design can be used to describe Learning Objects/Activities for the purpose of sharing and reuse of learning materials across different applications.

The architecture in Figure 4 lays out all components that are necessary to design and build context-aware mobile learning applications. In the next section, we will consolidate various ideas that we have so far discussed in this thesis by describing a conceptual IMS LD Level-C compliant context-aware mobile learning application.

3.3. Context-Aware Personalized Revision Aide (CAPRA)

In the previous section, we have proposed an architecture for context-aware mobile learning applications and described its components. In this section, we conceptualize an IMS LD Level-C compliant context-aware mobile learning application to consolidate our review so far. We call this application Context-aware Personalized Revision Aide (CAPRA). Here, we describe a brief application scenario for illustration purpose. In this scenario, CAPRA is used as an educational application in a University module with a class size of forty. Here, each student has a smart mobile device which was agreed and registered to use CAPRA for the module revision. The following sub-sections discuss the overview, functionalities, and architecture of CAPRA.

3.3.1. Overview

CAPRA is a personalized context-aware learning application aimed to make the revision of lecture materials easier and more efficient. CAPRA consists of three key components: (i) CAPRA learning app, (ii) CAPRA server and (iii) CAPRA resource editor. Figure 6 shows the general overview of the CAPRA system.

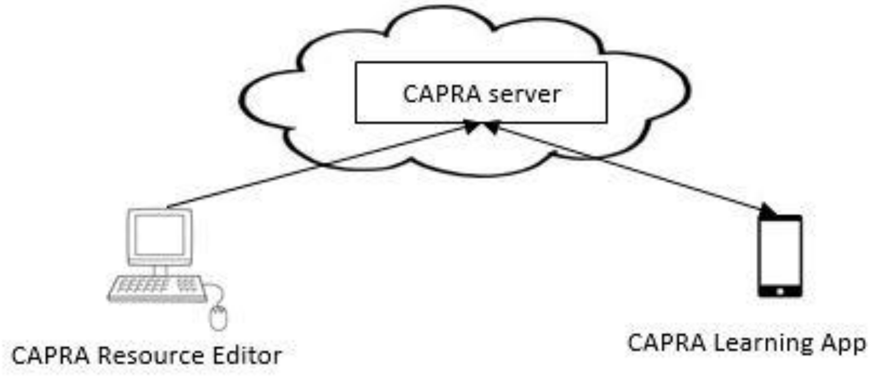


Figure 6: General overview of CAPRA system

The CAPRA resource editor is a client-side web application that allows lecturers to prepare learning materials and questions to be used in the CAPRA system. The output of the CAPRA resource editor is in the IMS LD format and stored in the CAPRA server. Since the learning materials are in the IMS LD format, these can be exported and re-used in systems other than CAPRA. The CAPRA server is the key component that oversees the whole operation of CAPRA system. It stores (i) learning resources, and (ii) Learner’s Context Model. The learning resources are generated by the CAPRA resource editor and also include questions to be used in assessing a learner’s knowledge. Based on the learning resources and the learner’s knowledge, the CAPRA server personalizes materials for each learner. The CAPRA server also provides resource adaptation based on the learner’s context. We will discuss more about the Learner’s Context Model in section 3.3.4, and personalization and context adaptation in section 3.3.5. The CAPRA learning app is a mobile application which is used by learners. It serves as the access point for learners to obtain the learning materials. It also acquires a learner’s context information and updates the CAPRA server.

3.3.2. Key functionalities of CAPRA

CAPRA has three main functionalities from the learner's view point: (i) assessing a learner's knowledge, (ii) recommending personalized and context-aware learning materials, and (iii) notifying the learner if there is a peer nearby who might be of help to the learner.

Figure 7 shows the general flow of the functionalities of CAPRA. When a learner starts the CAPRA learning app, it will show a list of available lessons to choose from. Once a learner chooses a lesson, CAPRA will generate a personalized quiz based on the chosen lesson and the learner's knowledge. After the learner finishes the quiz, CAPRA will check if the learner knowledge of any particular concept is weak. If so, CAPRA provides the relevant learning materials for the learner to review. Once the materials are reviewed, there will be another personalized quiz provided to the learner as shown in Figure 7. If CAPRA deems that the learner has adequate knowledge about every concept in the lesson, the lesson is over and the learner can choose another available lesson. The learner can exit the application at any step and continue with it at a later time as the session is saved.

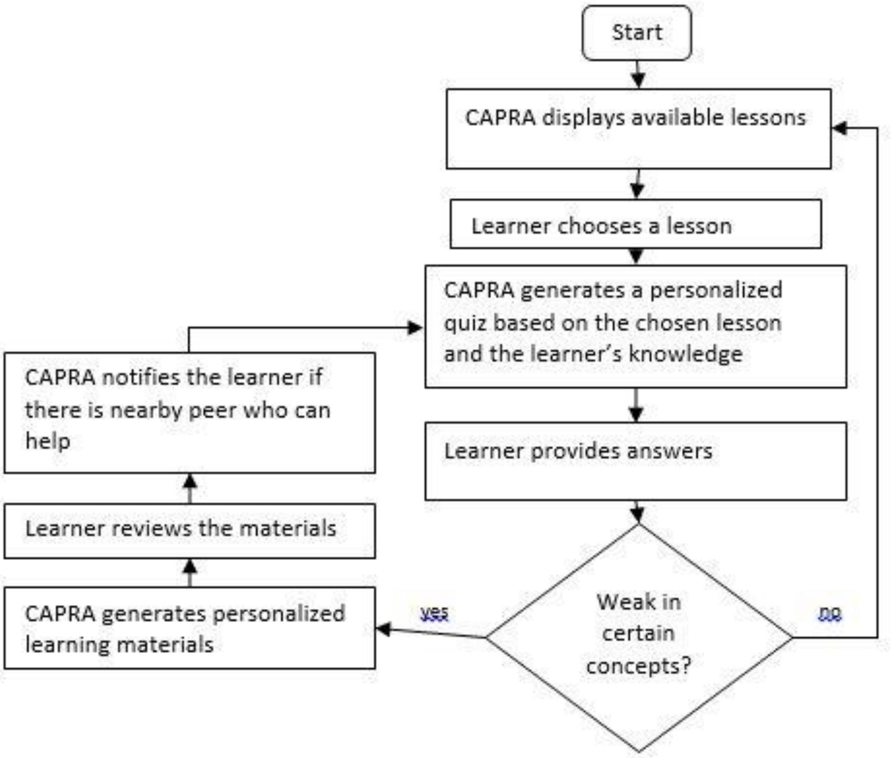

Start
CAPRA displays available lessons
Learner chooses a lesson
CAPRA generates a personalized quiz based on the chosen lesson and the learner's knowledge
Learner provides answers
Weak in certain concepts?
yes
no
CAPRA generates personalized learning materials
Learner reviews the materials
CAPRA notifies the learner if there is nearby peer who can help

Figure 7: Functionalities of CAPRA

The notification functionality of CAPRA depends on the learner's knowledge level, his or her current location, and the willingness of nearby peers to help. For example, John is a student taking the class and has done the revision of the Design Thinking lesson using CAPRA. CAPRA detects that he is particularly weak in understanding the concept "Ideate". As a result, CAPRA notifies him when there is a peer nearby who has a good understanding of the concept "Ideate" and who is willing to help so that John can approach him or her for discussion. A learner can configure the setting in the CAPRA learning app to "Do not disturb" if he or she is unwilling or unavailable to help others.

3.3.3. CAPRA's Architecture

CAPRA utilizes the general architecture of a context-aware application proposed in section 3.2. Figure 8 shows the architecture of CAPRA. The Personalization and Context Adaptation module has two main tasks i.e., generating quizzes and recommending learning materials. Both learning materials and quizzes are personalized and adapted based on the Learner's Context Model which in turn includes the Learner Model and Context Model. Lecturers can use the CAPRA Resource Editor to generate learning materials and quizzes in IMS LD format. These materials are then stored in the database in the CAPRA Server. The CAPRA Learning App not only acts as the access point for a learner but also acquires context information and updates the CAPRA server. The Notification module notifies a learner if there is a nearby peer who can help him or her in the revision. It uses the Learner Context's Model to decide whether or not to send a push notification to a learner about a nearby peer.

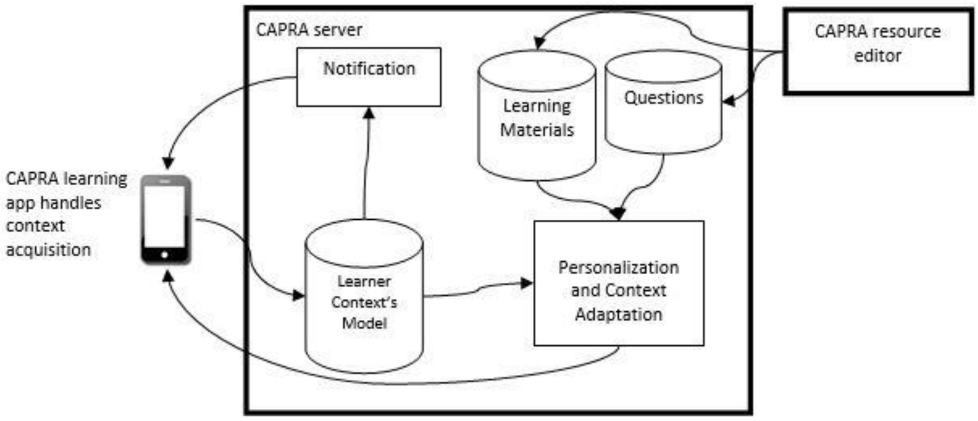


Figure 8: Architecture of CAPRA system

3.3.4. CAPRA’s Learner's Context Model

In this section, we discuss how we can model learners for the implementation of the CAPRA system. Since CAPRA is a personalized context-aware application, it's Learner's Context Model consists of both the Learner Model and Context Model information.

CAPRA's Learner Model includes learner’s personal information and knowledge. CAPRA's learner's knowledge model makes use of the overlay model (described in section 2.4.1) where each lesson is divided into concepts with each concept having a scale of 0 to 5 depending on the learner's expertise. 5 means a learner has a complete understanding of a concept while 0 means a learner has no knowledge or has not been assessed for a particular concept. For example, a lesson on Design Thinking may include 5 concepts - empathize, define, ideate, prototype and test. A learner who is working on the Design Thinking lesson will have one score for each concept in the “Design Thinking” lesson.

Table 3: Learner model of John

Personal information	**name**: John **email**: john@gmail.com **contact**: 9543 5940
Knowledge	**lesson**: Design Thinking **concepts**: Emphasize: 5 Define: 4 Ideate: 1 Prototype: 2 Test: 4

The Learner Model, for example, for a student called John who takes the Design Thinking lesson would capture information as shown in Table 3. In this case, John has good understanding about the concepts “Empathize”, “Define” and “Test” and needs more revision on “Ideate” and “Prototype”. He will be recommended learning materials related to “Ideate” and “Prototype” instead of the whole “Design Thinking” lesson.

Deciding which contextual information to use in an application depends on requirements of the application. In CAPRA's Context Model, we use 6 types of contextual information. They are location, peer nearby, device type, network connectivity, ambient light, and noise level. We need location information to derive the information on peer nearby because students need to find nearby peers for help with their revision. We also need the device type information to adapt learning resources to suit a student’s mobile device. Network connectivity information is also important in deciding whether a video or photo is sent to a student. Ambient light and noise level are needed to optimize lighting and audio level of videos and audio resources for a conducive condition for revision. Table 4 shows the context information and how they are used in the application. The context information is captured through the sensors of a learner’s mobile phone. Location of each learner is detected by GPS sensor on his or her phone and then uploaded to the CAPRA server. The CAPRA server then derives the People nearby using respective position coordinates. The CAPRA learning app also reports to the CAPRA server on the device type and network connectivity strength.

Table 4: Contextual information used in CAPRA

Context	**Active/ Passive**	**Primary/ Secondary**	**Possible Values**	**How context is used**
Location	Passive	Primary	Coordinates	Location is used in deriving the information of peers nearby.
People nearby	Active	Secondary	List of learners	A learner is prompted if there is a peer nearby from whom he can get help on revising certain concepts. A nearby peer is qualified to help if his or her knowledge model shows better understating of a particular concept than the learner.
Device type	Active	Primary	Models of mobile devices	Device type is used to adapt learning material to a suitable format. For example, the layout and font size of the reading material may be adapted to suit the device that is being used.
Network connectivity	Active	Primary	High, Average, Low	Network activity is used to adapt learning material to avoid disruption. For example, if the network connectivity is low, instead of sending a video resource, the system will send an audio version to avoid a scenario where a learner has to wait for a long time to load the learning materials.
Ambient light	Active	Primary	High, Average, Low	The light level of the screen will depend on the ambient light to provide most conducive viewing.
Noise level	Active	Primary	High, Average, Low	Noise level is used to determine which type of learning material to send. For example, if the surrounding noise level is high, the system will set the sound level to maximum.

3.3.5.CAPRA's Personalization and Context Adaptation

CAPRA uses the overlay model (described in section 2.4.1) for the learner’s knowledge model. As a result, personalization is possible at the concept level instead of lesson level. Devising a complex algorithm for personalization and context adaptation is not a purpose of this paper. Therefore, we utilize simplified algorithms in the personalization and context adaptation process of CAPRA. To personalize the quiz, we select the concepts for a chosen lesson with low knowledge level (knowledge level of 3 and below) for a particular learner. We then choose questions which are related to the selected concepts. In CAPRA, we assume that all the questions are of the same difficulty level. However, a more complex system can be implemented by assigning difficulty levels to each question and utilizing this information in updating learner's knowledge model and choosing questions. CAPRA also keeps track of the questions that a particular learner has already tried and avoids selecting those questions if he or she has already got them correct. Similarly, in recommending learning materials, the materials are selected based on a learner's knowledge model.

CAPRA utilizes context information in two different ways. The first is to adapt learning materials before sending them to the CAPRA mobile app. The adaptation is based on device type and network connectivity. Based on the device type, CAPRA server gets the screen dimension of the learner's device and adapts the resource images and contents to suitable sizes and fonts. Network connectivity decides the format of the learning resources that the CAPRA server sends to the CAPRA learning app. If the network connectivity is high, CAPRA can send a high quality video resource while if it is low, it may send a low quality audio file. The second way in which the CAPRA server utilizes the context information is in the notification of the learner when there is a nearby peer who can help him or her in the revision. The CAPRA server finds the nearby peers of a particular learner by using their current locations. If learners are within 400m distance from each other, they are considered to be nearby. The CAPRA learning App also utilizes the context information. It uses ambient light and noise level to provide a better learning experience. If the ambient light is low, the CAPRA learning app will increase the screen brightness of the device and vice versa. In the same way, the CAPRA learning app will increase the sound volume if it detects high noise level in the surrounding.

3.4. Summary

In this chapter, we have proposed a refined general architecture for context-aware mobile learning applications and discussed each component of the architecture – namely (i) Learner Model, (ii) Context Acquisition, (iii) Context Model, (iv) Learning Objects/Activities, (v) Personalization, (vi) Context Adaptation, (vii) Learning Objects/Activities Editor, and (viii) Notification. We have also conceptualized a context-aware mobile learning application called CAPRA to consolidate our discussion so far in this thesis. From our discussion above, we have seen how various components of the whole system work together and what are the key considerations for implementing context-aware mobile learning application. In the next chapter, we consider context-aware applications in classroom scenarios.

Chapter 4 Review on Context-aware Applications in Classrooms

4.1. Introduction

In chapter 2, we have reviewed the state of arts in E-learning applications. In the chapter 3, we have refined and extended a general architecture for context-aware mobile learning applications and conceptualized a context-aware application, CAPRA, to consolidate our review and ideas. However, learning applications are not the only applications which benefit from context information. This information can also be used in classrooms to assist in classroom activities. In this chapter, our main contribution is to review the state of the art in Context-aware applications in classrooms and identify the challenges in developing such applications. This chapter is structured as followed.

- Section 4.2 explores the existing classroom Context-aware systems
- Section 4.3 discusses the various scenarios where Context-aware information can assists in classroom activities
- Section 4.4 discusses important processes in the classroom Context-aware application
- Section 4.5 discusses the challenges of developing classroom Context-aware applications
- Section 4.6 summarizes the chapter

4.2. Existing classroom context-aware systems

There are many existing classroom applications which utilize varying degree of context information. Some applications use explicit context information specifically defined by an instructor while others use implicit context information. In this section, we discuss a few proposed systems by various researchers.

Anderson, et al. (2007) proposed a Tablet PC based classroom system, Classroom Presenter, to integrate activities into the lecture. The presenter allows students and lecturers to share their slides among each other and also show on the big screen simultaneously, enabling students to work on the examples that a lecturer is going through in a class and share their attempts with the lecturer or the class. The system prototype was tested in a senior level Algorithms course in a public University in the United States. The students are asked to solve example problems during the lectures and to send in their responses as they finish. This enables the instructor to check students' effort and gauge the class understanding of the lesson immediately. The system also allows lecturers to show students' answer on the screen anonymously for the discussion purpose. In this system, the context is the example problem that is currently in discussion during the lecture and it is explicitly chosen by the lecturer. Although, Anderson, et al. (2007) did not evaluate the effectiveness of the system on learning outcomes formally, the classroom activities appeared successful in helping students understand problems better during the lecture.

Shi, et al. (2003) proposed the Smart Classroom project which closes the gap between tele-education and traditional class-room activities by utilizing smart space technologies in a real classroom. The purpose is to carry out the class where students are in multiple different physical locations. The classroom setup involves two wall-size projector screens and one of them is a touch-sensitive SmartBoard screen. The remote students can view the lesson through a client program. Multiple cameras are also set up in the classroom to capture the live video and the teacher wears wireless microphone to capture his or her speech. The system also provides interaction between remote students and the teacher or the local students. A few noble input methods are also used in the system. For example, Speech-capable virtual assistant allows the teacher to command common tasks using voice. The biometrics-based login using face-recognition and speak-verification technologies are also utilized to authenticate a teacher before the use of the classroom facilities. Smart cameraman also follows the flow of the class depending on the context of the situation. For example, if a teacher is writing on the board, the system will capture the board. If a teacher is holding and showing a model, the camera will automatically zoom in on it. Shi, et al. (2003) identifies that the infrastructure should provide a high-level abstraction and coordination mechanism for different components to work together in a system like the Smart Classroom. Also, they mentions that it is important for different part of the systems to be able to exchange data at real time as they need latest data from one another to fulfill their tasks.

Based on the Smart Classroom project by Shi, et al. (2003), Suo, Miyata, Morikawa and Ishida (2009) proposed the open Smart Classroom to enable extensibility and scalability of the system. They argue that allowing remote students to communicate with the teacher and the local student is inadequate and the system should allow two or more real-time interactive classrooms to connect with each other. Two Open Smart Classrooms were built – one at the Tsinghua University and another at the Kyoto University. The two classrooms are then connected by internet and controlled by the Open Smart Platform to communicate and coordinate with each other. The classroom at the Tsinghua University is attended by Chinese speaking students while the one at the Kyoto University is attended by Japanese speaking students. The class is then conducted in English and the Open Smart Platform provides the translations through web services for both classes. The live video in both classes shows appropriate content depending on the context of the class.

As we conduct the review, we realize that there are only a few existing classroom context-aware applications and their practical use are still limited. One reason might be due the technical complexity involved in developing such applications. There is also a lack of frameworks which allows easy and rapid development. To solve this problem, we propose and develop SCF which acts as a foundation for developing classroom context-aware applications in Chapter 5. Before going into the details about the framework, we first discuss about various classroom activities which might benefit from the use of context information in the next section.

4.3. Classroom activities

In this section, we discuss various scenarios where context information can assist in classroom activities.

4.3.1. Attendance taking

One of the most common classroom activities is the attendance taking activity. The key task is to identify the students who are present during a class. The task involves three questions – “who”, “when” and “where”. “Who” concerns with the identity of a student while “when” concerns with the classroom time and “where” concerns with the location of the classroom. Various methods can be used to take attendance such as manual method with paper and pen, barcode scanner (Schneider, 2010) or finger print scanner (Taxila, 2009). For example, finger print scanners can be fitted to each classroom and students can scan their fingers while the connected computer records the attendance. However, it is still not fully automated as it requires explicit input (scanning finger) from students. A context-aware application with identification capability can automate the attendance taking activity.

4.3.2. Classroom notice and announcement

In addition to questions about “who”, “when” and “where” from the attendance taking scenario, this scenario also involves about “what”. “What” concerns about what information needs to be made known to students. Traditionally, an instructor may make an announcement verbally during the class. However, if the announcement is lengthy and not crucial for the learning activity, it uses up precious time unnecessarily. A context-aware application which can automatically send notices and announcements to students in the class can improve the efficiency greatly.

4.3.3. Classroom poll or quiz

This scenario requires explicit user input from students. Traditionally, an instructor may conduct a classroom poll or quiz manually in the class. However, as with the previous scenario, this can use up precious classroom time.

4.3.4. Sharing digital learning materials

Many higher education students download lecture notes, videos and other learning resources from school systems such as Integrate Virtual Learning Environment. The key task is to handle the transfer of files to students' computers. However, this process can be automated through the use of a context-aware application in a class. For example, when a student walks into a classroom, the application detect his presence and automatically trigger the download of learning resources for the class into his mobile phone or laptop.

4.3.5. Classroom presentation

Students sometimes have to prepare PowerPoints slides or visual aides to present their ideas or work to the class. A significant amount of classroom time is spent of setting up their slides on the classroom computer. It will save time if a context-aware application can automatically set up the slides for students. For example, Bravo et al (2007) proposes the use of RFID technology to identify students and display appropriate content accordingly. They have developed a tool for students to create a visual presentation with text and images and store it on the classroom computer. When a student comes near the board for the presentation, the reader on the board detects the student’s RFID label and displays his or her presentation content.

4.4. Important processes classroom Context-aware applications

4.4.1. Student Identification Process

In the previous section, we have discussed various classroom activities which can benefit from the use of context-aware applications. From the discussion, we can see that student identification process is the foundation of context-aware applications in classroom. In all situations, we need identify a student in a class automatically. Various methods can be used to achieve this.

Bravo, Hervas and Chavira (2005) propose the use of Radio Frequency Identification (RFID) to identify students without any user interaction. With the use of RFID, three different parts are involved - the reader, the label (the tag) and the computer. The readers are fixed to classroom and are connected to computers while students carry around the labels which contains information which uniquely identifies the student. When a student gets near to a reader, the reader automatically detects the label, extract information from it and identify the student. Bravo, Hervas and Chavira (2005) argues that the use of RFID is better than traditional barcode because the labels do not have to be visible to be read and the reader can be located a meter away. Also, in RFID, the information is encrypted before transmission and only authorized readers can update the data. Lim, Sim and Mansor (2009) also propose the use of RFID for the attendance taking purpose.

With the increasing use of smart mobile devices by students, Bluetooth technology count be used for the student identification process. Students can be identified from their phone Bluetooth address instead of the RFID label. A classroom can be equipped with a Bluetooth device which can periodically scan the nearby Bluetooth devices and identify students. Figure 9 shows a possible classroom set up for student identification in a class. The classroom server has the Bluetooth capability and handles the student identification.

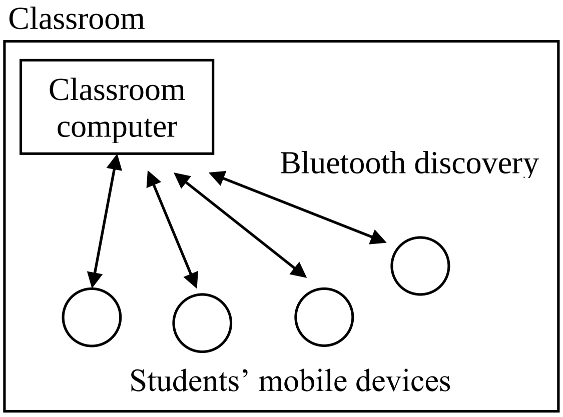


Figure 9: Classroom setting

4.4.2. Data Communication

Classroom context-aware applications also need to send and receive data from students' devices. For example, in the classroom poll scenario, the application needs to send poll data to a student's device and receive his or her response to the poll. The Bluetooth connection can also be used to communicate poll data. However, using Bluetooth has its limitations. For example, Bluetooth devices allow limited number of concurrent connections. As a result, it will not work if a lot of students are taking the poll at the same time. Also, scenarios such as sharing digital learning materials requires the transfer of sizable files from the application to a student's device and the use of Bluetooth connection will not be suitable.

One solution to such problem is to use normal HTTP protocol. As a result, the classroom context-aware application needs to have a HTTP server and web services. We propose to use the same set up as shown in Figure 9 and host the HTTP server on the classroom computer. The student identification process can be done through the Bluetooth connection as discussed in the last section. Once a student is identified and verified, the classroom server can send the IP address of it HTTP server to a student's device through the Bluetooth connection. Once the student's device know the IP address of the HTTP server on the classroom computer, it can communicate with it through normal web services.

4.5. Challenges

As we have seen in the previous sections, a context-aware application can be helpful in various classroom activities and needs to be able to handle various scenarios. As such, instead of viewing the classroom context-aware application as one system, it is more suitable to view it as a suite of applications working together to fulfill various needs of different classroom activities. It also makes the system more extensible from the architecture point of view.

Figure 10 shows a possible setup for a context-aware system in a university. The classroom servers are then used to connect to students’ devices to carry out activities such as taking attendance and carrying out classroom polls while the central server coordinates classroom servers.

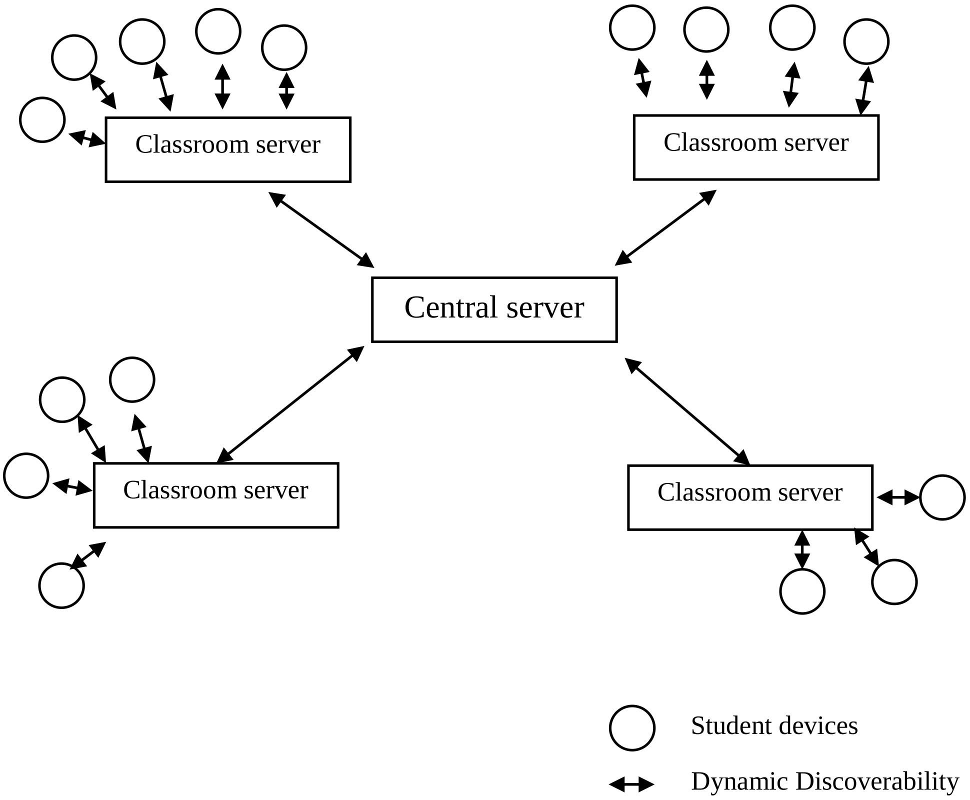


Figure 10: A possible setting for a context-aware system in a university

In implementing a system as shown in Figure 10, we have identified three key challenges as follow.

- Structure – As we have discussed previously, it is better to view the system as a suites of applications working tougher than a large system. For example, the system in the Figure 10 involves the central server, classroom servers and student devices. Even on the one classroom server, it might involve various applications which assists in different classroom activities. A proper software structure is needed to allow these various components and applications to interact with one another seamlessly.

- Extensibility – Education institutions or even different instructors from the same institution have different needs for teaching and learning support. New learning activities may require new functionalities. It is therefore important to have an extensible system which allows the addition of new modules and parts to the existing system.
- Scalability – with any large systems, scalability is an important issue. For example, it is important to have a mechanism to handle added load from increased number of users.

4.6. Summary

In this chapter, we have explored the use of context-aware applications in classrooms. We first discussed about various classroom activities which can benefit from the use of context-aware applications such as attendance taking, classroom notice and announcement, classroom poll and quiz, sharing digital learning materials and classroom presentation. We have also discussed about the student identification process and various way the data can be communicated between a context-aware application and students' devices. Moreover, we have also identified three key challenges in building a classroom context-aware application. In the next chapter, we propose Smart Classroom Framework (SCF) to solve the identified challenges and provide a foundation for building classroom context-aware applications.

Chapter 5 Smart Classroom Framework

5.1. Introduction

In the previous chapter, we have discussed about the use of context-aware applications in classrooms and the challenges involved. In this chapter, our main contribution is to propose and build the Smart Classroom Framework (SCF) to overcome the challenges identified in section 4.5 and provide a foundation for building a classroom context-aware system. This chapter is structured as follow.

- Section 5.2 explains the overview of SCF
- Section 5.3 discusses SCCentral
- Section 5.4 discusses SCHub
- Section 5.5 discusses SCApp
- Section 5.6 discusses SCStudentApp
- Section 5.7 highlights technical challenges and summarizes the chapter

5.2. Smart Classroom Framework

5.2.1. Overview

In this section, we propose Smart Classroom Framework (SCF) to provide the foundation for building large-scale context-aware systems for educational purposes. It is a light weight framework and provides a solution to the three challenges – (i) Structure, (ii) Extensibility and (iii) Scalability, that we have discussed in Section 4.5.

Structure

SCF provides a structure for context-aware systems with four main components – (1) SCCentral, (2) SCHub, (3) SCApps and (4) SCStudentApp.

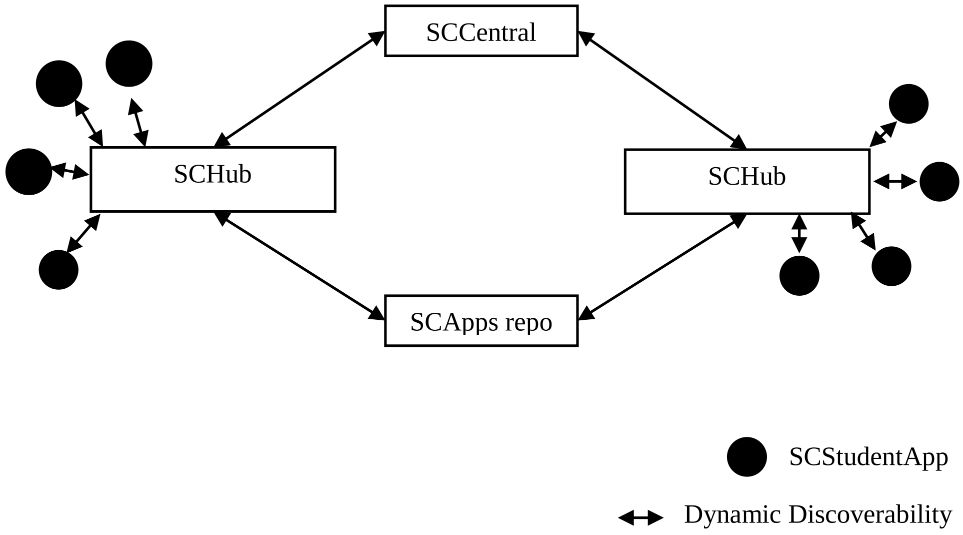


Figure 11: Overall architecture of SCF

Figure 11 shows the overall architecture of SCF with two SCHubs in operation. The SCCentral is the software that runs on the central server from Figure 10. It is the glue that holds the whole system. We assume that SCCentral is on a university's server with a known IP address. The SCHub is the software for classroom servers. A raspberry pi can be used as a portable classroom server to house the SCHub software. When a SCHub is online, it registers its IP address with the SCCentral through the HTTP protocol. Students use the SCStudentApp to access SCHubs. SCApps are installable packages that can be installed to SCHubs. For example, Smart Attendance SCApp extends SCF and assists instructors in taking attendance during classes. SCApps are available in the SCApps repository for downloads.

Extensibility

In SCF, The use of SCApps allows the extensibility to the SCF. SCApps can be installed to the SCHubs on the go using the SCHub Administrative portal through a normal web browser. The extensibility allows new functionalities, use cases and scenarios to a SCF system.

Scalability

SCF allows the use of as many SCHubs in the system. Since SCHubs operate mostly independently, they are not affected by the number of SCHubs in the system.

5.2.2. Stakeholders

Before we go into details on how SCF works in the next sections, we discuss about different users groups who are involved in using SCF. They are

- System administrators
- System developers
- Instructors and
- Students

System administrators are responsible for managing a SCF system. They use the SCCentral Admin portal to carrying out administrative tasks such as managing user accounts for instructors and students, setting up and registering new SCHubs and rolling out new SCApps. SCApps are implemented by system developers to extend the functionalities of SCF. Instructors use SCHubs as teaching support tools in their classroom activities. They use the SCHub portal to install and use SCApps easily as and when necessary. Students use SCStudentApp to access to SCHubs.

5.2.3. Technologies

SCF is built on top of various tried and tested technologies. The followings are the main technologies involved in the implementation of SCF.

- Programming Language – We use Python as the main programming language for the implementation of SCF. Alternative languages that we have considered are Ruby and PHP. PHP is known to be hard to maintain and non-modular. With the advancement of frameworks such as Symfony and Lavaral, the situation with PHP has improved. However, compared to other languages such as Ruby and Python, PHP lags behind in the improvement of its development tools. Ruby is a beautiful language with very readable syntax and is a close second behind Python in our choice of language for SCF. We have chosen Python as it is a widely used general-purpose, high-level programming language with supports for various programming paradigms, such as functional, imperative, object-oriented or procedural styles. It is also popular among research community and has a very strong community supports and has a wide array of open-source libraries and packages.

- Framework – With the choice of Python as the main programing language, Django becomes our choice of framework as it is a de facto framework for python when it comes to web-related development. Django focuses on rapid development and clean, pragmatic design. It provides object-rational mapper, automatic admin interface, elegant URL design, template system, cache system and internationalization out of the box. Since Django is based on python, other python libraries can also be used within the system. Django is chosen due to its strong and active community support.
- Databases – both SQLite and MySQL are used for in SCF. SQLite is a software library that implements a self-contained, server less, zero-configuration, transactional SQL database engine. Thus, it is suitable for the use on SCHub. SQLite is also the most widely used SQL database in the world. On the other hand, MySQL is a proven database management system for large-scale applications and is suitable for the SCCentral which may contain large amount of data to coordinate all SCHubs in the system.
- Bluetooth – Bluetooth is a wireless technology standard for exchanging data over short distances. It is managed by the Bluetooth Special Interest Group, which has more than 20000 member companies.

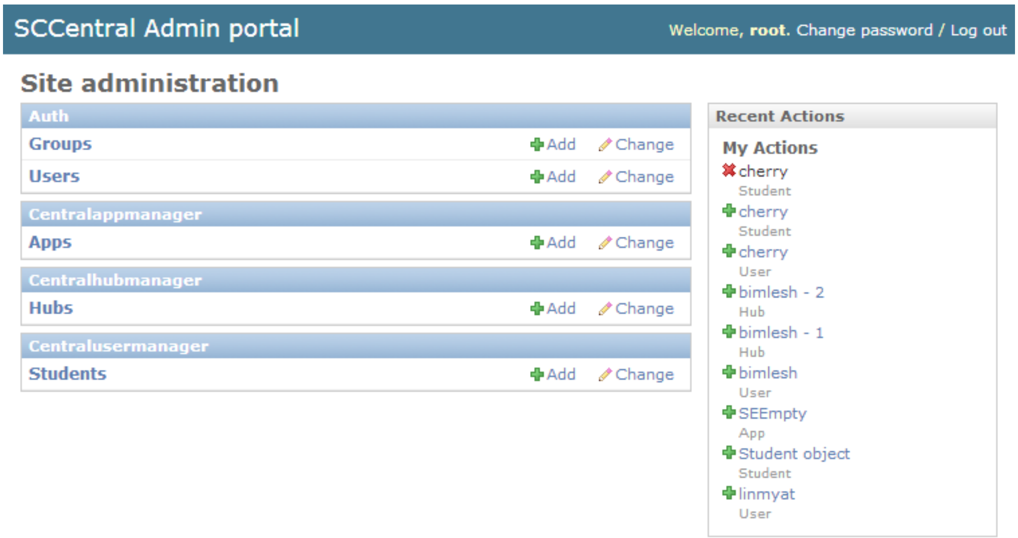
5.3. SCCentral

5.3.1. Overview

In the previous section, we have discussed the overview of SCF. In this section, we discuss about SCCentral in detail. SCCentral has three main responsibilities as follow.

User management

SCCentral allows system administrators to manage users' accounts for lecturers and students. It also provides functionalities for SCHub to authenticate students and retrieve their information. Figure 12 shows the Administrative Portal of SCCentral.



SCCentral Admin portal

Welcome, **root**. Change password / Log out

Site administration

Auth		
Groups	✚Add	✎Change
Users	✚Add	✎Change
Centralappmanager		
Apps	✚Add	✎Change
Centralhubmanager		
Hubs	✚Add	✎Change
Centralusermanager		
Students	✚Add	✎Change

Recent Actions

My Actions

✖cherry
Student

✚cherry
Student

✚cherry
User

✚bimlesh - 2
Hub

✚bimlesh - 1
Hub

✚bimlesh
User

✚SEEmpty
App

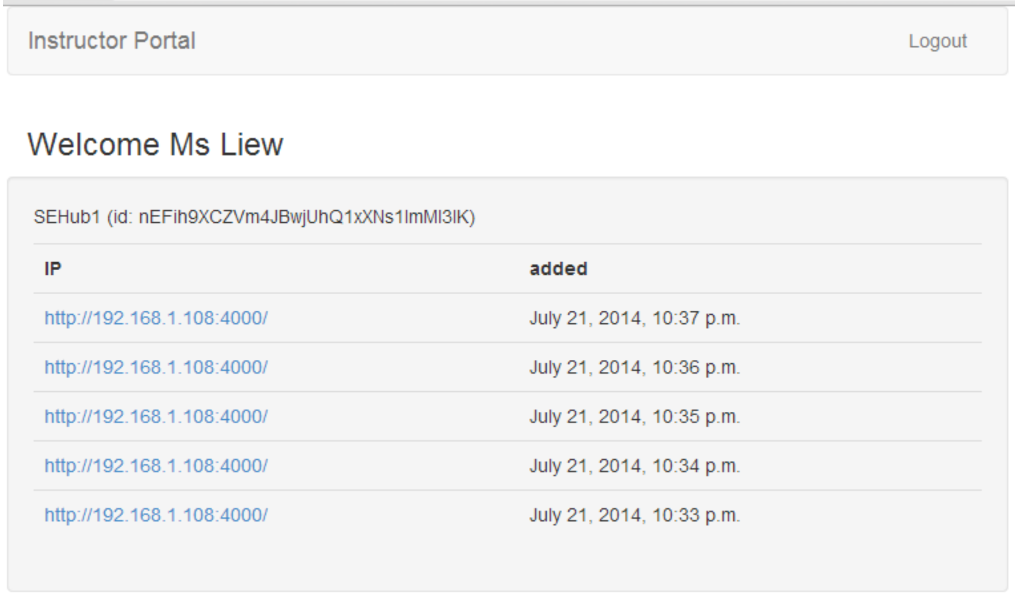
✚Student object
Student

✚linmyat
User

Figure 12: SCCentral Administrative Portal

Hub management

SCCentral also keeps track of SCHubs in the system. For example, it keeps track of the assignment of SCHubs to instructors. It also allows a SCHub to update its current IP address and status. Moreover, it allows instructors to log into the Instructor Portal to access their SCHubs through a web browser. Figure 13 shows the Instructor Portal. The instructors use the Instructor Portal to find out the current IP address of their SCHubs.



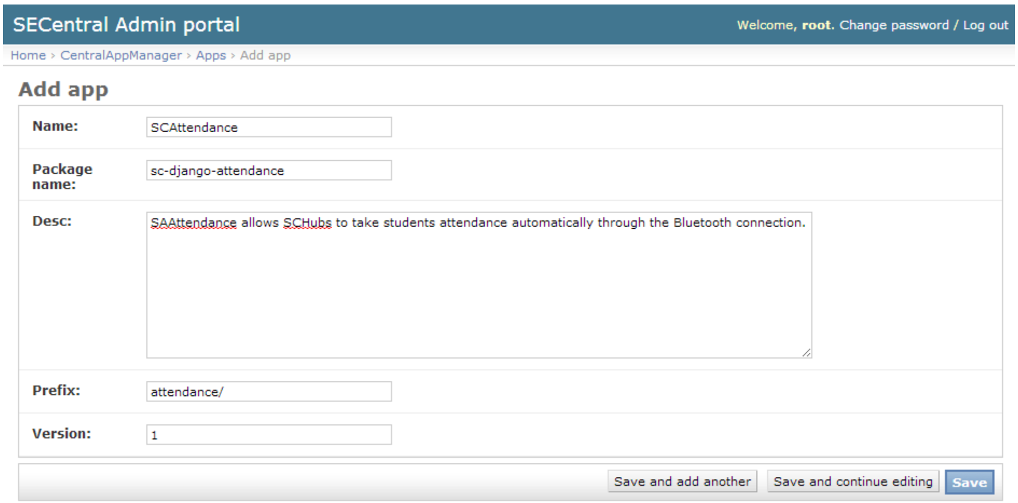
Welcome Ms Liew

SEHub1 (id: nEFih9XCZVm4JBwjUhQ1xXNs1ImMI3IK)

IP	added
http://192.168.1.108:4000/	July 21, 2014, 10:37 p.m.
http://192.168.1.108:4000/	July 21, 2014, 10:36 p.m.
http://192.168.1.108:4000/	July 21, 2014, 10:35 p.m.
http://192.168.1.108:4000/	July 21, 2014, 10:34 p.m.
http://192.168.1.108:4000/	July 21, 2014, 10:33 p.m.

Figure 13: Instructor Portal in SCCentral

App management



SECentral Admin portal
Welcome, root. Change password / Log out
Home › CentralAppManager › Apps › Add app

Add app

Name:	SCAttendance
Package name:	sc-django-attendance
Desc:	SAAttendance allows SCHubs to take students attendance automatically through the Bluetooth connection.
Prefix:	attendance/
Version:	1

Save and add another
Save and continue editing
Save

Figure 14: Adding a new SCApp information

SCCentral also keeps track of available SCApps in the SCApp repository. We will discuss more about SCApp in section 5.5. Figure 14 shows a screenshot of adding a new SCApp Information.

5.3.2. Implementation Details

SCCentral is implemented as a Django project and is dependent on various third-party. The followings are the third-party applications used in the SCCentral.

- django.contribe.contenttypes – This application tracks all the models installed in the Django-powered project and provides a high-level, generic interface for working with the models.
- django.contribe.sessions – This application allows the storage and retrieval of arbitrary data on a per-site-visitor basic. It stores the data on the server and handles the sending and receiving of cookies.
- django.contrib.auth – This application handles authentication and authorization. Authentication verifies a user and authorization determines what an authenticated user is allowed to do.
- django.contribe.messages – This application allows the display of a one-time notification message to the user after processing user input.
- django.contrib.admin – This application provides the admin interface for the SCCentral administrative portal.
- rest_framwork – This application is a third-party application which allows an easy implementation of RESTFUL web services

The source code of SCCentral is available online at https://bitbucket.org/soelinmyat/sccentral/.

5.4. SCHub

5.4.1. Overview

SCHub acts as a classroom server. SCHub can be installed on a laptop, desktop or any other machine with a UNIX based operating system and Bluetooth capability. It interacts with SCStudentApp to assist in teaching and learning activities. It has four main responsibilities as follow.

Bluetooth management

The SCHub software provides functionalities to discover nearby Bluetooth devices and advertise Bluetooth services to share the SCHub IP address with nearby IP address. The following steps show the discovery process of a nearby SCHub IP address.

- A student's devise discover nearby Bluetooth devices
- The student's device check for services with UUID "720b0190-7e91-4a51-a4ac-000000000001". A universally unique identifier (UUID) is an identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). We use the above UUID string to identify SCF related Bluetooth services.
- The student's device connects to the service and sends its username and keypass. A keypass is a secret key that is generated by the SCCentral when a student log in to the SCCentral from the SCStudentApp. The use of the keypass allows SCHubs to validate students without knowing their passwords.

- SCHub check username and keypass against the SCCentral database to see if they are valid. When communicating with the SCCentral server, the SCHub needs to provide its hub id and hub secret key as a security measure.
- If the user credentials are valid, SCHub send back its IP address to the student's device.

The above process requires a student's device to connect to the SCHub and exchange data (username, keypass) with the SCHub. In practice, we have found that it is not necessary. Alternatively, we can use the discovery process below to simplify the steps.

- When a SCHub is powered on, it sets its Bluetooth device name as "SCHub-http://ip_address_of_the_hub/"
- A student's devise discover nearby Bluetooth devices
- The student's device checks the device names to see if "SCHub" substring is included. If it is, the student's device extracts the IP address from the name.

The second process only requires students' devices to check the name of the nearby devices and does not require direct Bluetooth communication between students' devices and the SCHub. Also, since the keypass and hub secret are used, we do not need to worry about malicious devices, pretending as a SCHub.

Both discovery processes are supported by the SCF out of the box and system administrators can set up SCHubs in a way they want to use. However, at the current version, only the local IP address of the server is provided. It means that the mobile device and the SCHub must be in the same local Internet connection to communicate with one another. It is not a problem in a scenario where both mobile devices and the SCHubs are connected to a school wireless network with school routers set for port forwarding to the SCHubs.

RESTFul API management

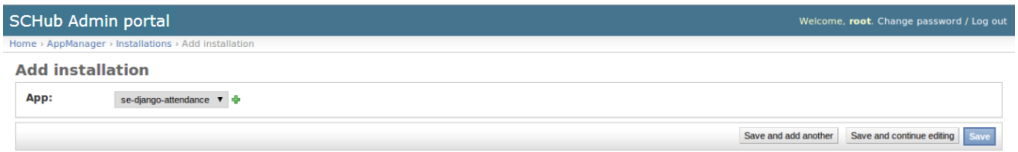
SCHub is also responsible for providing a RESTFul server which can be extensible by SCApps. When a SCApp is installed, the available RESTFul APIs of the SCHub is extended dynamically. For example, the Smart Attendance Application has a prefix “attendance” and a URL extension of “take” to allow attendance taking. When the Smart Attendance Application is installed, “http://ip_address_of_hub/attendance/take” becomes available. We describe the installation process of SCApp in the next sub section.

SCApp management

SCHub provides an easy way for instructors to install SCApps and use them as and when necessary. The following describes the installation process of a new SCApp to the SCHub.

- Instructor logs in to SCHub Administrative Portal and choose a SCApp to install as shown in Figure 15
- SCHub uses a tool called PIP to download the selected SCApp from Pipy and installs it to SCHub
- SCHub adds the selected SCApp to the installed app list
- SCHub adds the url extension of the SCApp to the installed url list
- SCHub creates necessary tables in the database for the new SCApp

- SCHub restarts the server


SCHub Admin portal
Welcome, root. Change password / Log out
Home › AppManager › Installations › Add installation
Save and add another
Save and continue editing
Save

Add installation

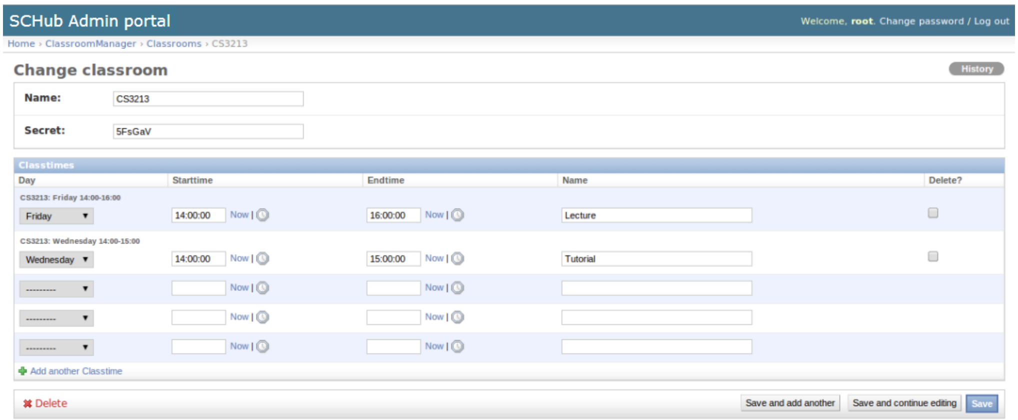
App: se-django-attendance ▼

Figure 15: Screenshot of Installation of SCApp

In a similar way, instructors can also use the SCHub Administrative Portal to uninstall SCApps from the SCHub.

Classroom management

SCHub also provide an easy way for instructors to manage classroom information and the enrolment of students to a class. A class can have multiple classroom times (e.g. lecture time, tutorial time) and students can be enrolled to a class. Figure 16 shows a classroom page in the SCHub Administrative Portal.


SCHub Admin portal
Welcome, root. Change password / Log out
Home › ClassroomManager › Classrooms › CS3213
History
Add another Classtime
Delete
Save and add another
Save and continue editing
Save

Change classroom

Name: CS3213

Secret: 5FsGaV

Classtimes

Day	Starttime	Endtime	Name	Delete?
CS3213: Friday 14:00-16:00 Friday ▼	14:00:00 Now |	16:00:00 Now |	Lecture	□
CS3213: Wednesday 14:00-15:00 Wednesday ▼	14:00:00 Now |	15:00:00 Now |	Tutorial	□
--------- ▼	Now |	Now |		
--------- ▼	Now |	Now |		
--------- ▼	Now |	Now |		

Figure 16: A classroom page in the SCHub Administrative Portal

5.4.2. Implementation Details

In the thesis, we propose to use a Raspberry Pi with the SCHub software as a class server due to its low price and mobility. We use a Raspberry Pi Model B with Rasbian operating system during the testing of the SCHub software. Like SCCentral, SCHub is also a Django project. In addition to the third party applications that are used in SCCentral, SCHub also uses the following applications.

- django_roa – This application allows a model to access from a remote data source. Each time a request is passed to the database, an HTTP request is made to the remote server with the right method (GET, POST, PUT or DELETE)

The source code of SCHub is available online at https://bitbucket.org/soelinmyat/schub/.

5.5. SCApp

5.5.1. Overviews

SCApps are installable Django applications which allow the extension of SCF functionalities. We use Python Package Index (PyPi) as the repository for SCApps. PyPi is a popular repository of software for the Python programming language with 46420 packages currently. Anyone can create a Pypi account for free and upload new software packages. We have discussed the installation process of SCApp into SCHub in section 5.4. We have implemented two SCApps as examples. They are Smart Attendance application and Classroom Poll application. Smart Attendance Application allows SCHubs to take students attendance automatically while Classroom Poll Application allows instructors to carry out classroom polls.

5.6. SCStudentApp

The SCStudentApp is a mobile application used by students to communicate with SCHubs. It has three basic responsibilities.

- Bluetooth management – SCStudentApp needs to be able to discover nearby SCHubs and communicate with it through Bluetooth connection.
- RESTFul API call management – SCStudentApp needs to be able to call RESTFul APIs on SCHub servers.
- Authentication – SCStudentApp needs to be able to authenticate students with the SCCentral server.

In addition, SCStudentApp also needs to supports functionalities required by the SCApps used in the system. For the testing purpose of SCF, we have implemented an Android application as a SCStudentApp.

The source code of the SCStudentApp prototype is available at https://bitbucket.org/soelinmyat/scstudentapp.

5.7. Technical Challenges and Summary

As a SCF system scales up, one potential challenge might be to optimize the SCCentral database. The database may need to be sharded and tables properly indexed to handle the heavy load. Also, SCStudentApp needs to be sensible in using the Bluetooth discoverability to preserve the battery of students' devices. Bluetooth Low Energy should be used instead of normal Bluetooth if a student's device supports it. A detailed experiment should be conducted to find an optimal condition for using energy intensive functionalities such as Bluetooth discovery. In addition, unstable connection between SCHub and SCStudentApp may post challenges in transferring or streaming large files between them.

In summary, we have proposed Smart Classroom Framework (SCF) which provides the foundation for the development and deployment of classroom context-aware applications. SCF provides a structure for the application with four main components – namely SCCentral, SCHub, SCApps and SCStudentApp. SCCentral coordinates SCHubs in the system and SCHubs are used for communication with SCStudentApp on students' mobile devices. SCF also provides the Bluetooth discoverability ability for SCHubs out of the box. Moreover, the use of installable SCApps allows the extension of SCF functionalities to cater to new use cases and scenarios. SCF is also scalable as SCHubs operate mostly independently and are not affected by the number of SCHubs in the system.

Chapter 6 Conclusion

6.1. Summary

In this thesis, we have explored the use of context-aware applications in the learning environments.

In Chapter 2, we have reviewed the state of the art in e-learning applications. We have reviewed various ways to classify e-learning applications and the definition of context. We have also reviewed the literature on learner modeling, Learning Design and various ways to carry out personalization and context adaptation.

In Chapter 3, we have proposed an architecture for implementing context-aware mobile applications. We have also consolidated our review and ideas by describing a conceptual context-aware mobile learning application called CAPRA.

In Chapter 4, we have discussed the use of context-aware applications in classrooms. We have discussed various classroom activities which can benefit from the use of context-aware applications such as attendance taking, classroom notice and announcement, classroom poll and quiz, sharing digital learning materials and classroom presentation. We have also discussed about the student identification process and various way the data can be communicated between a context-aware application and students' devices. Moreover, we have also identified three key challenges in building a classroom context-aware application.

In Chapter 5, we have proposed and developed the Smart Classroom Framework (SCF) which provides the foundation for the development and deployment of classroom context-aware applications. SCF provides a structure for the application with four main components – namely SCCentral, SCHub, SCApps and SCStudentApp. SCCentral coordinates SCHubs in the system and SCHubs are used to communication with SCStudentApp on students' mobile devices. Moreover, we have discussed the technical details of each component. We have also developed two SCApps, Smart Attendance application and Classroom Poll application, to show the robustness of the SCF.

6.2. Future Directions

From the discussion in section 2.4, it can be seen that there are many variations of the Learner Model. A promising future research direction is to standardize Learner's Context Model. There have been recent attempts by researchers (e.g., Das, et al. 2010, Zervas, et al. 2011) but there is still no standard which is commonly accepted. To standardize Learner's Context Model, we need to standardize both the Learner Model and Context Model. The key component of the Learner Model is the knowledge model which specifies the knowledge level of learners. It involves specifying a leaner's knowledge for a set of concepts. This naturally requires us to have a set of predefined concepts for learners to be evaluated on. This requires educational institutions and individuals to follow a particular education standard such as ABET (ABET) and National Council of Examiners for Engineering and Surveying (NCEES). It is also important to reconcile the differences between these standards to have a standardized knowledge model. A related research objective would be to standardize the Context model and extend Learning Design such as IMS-LD to include it. The outcome of such research will promote interoperability between different learning applications and ease the design and development of context-aware mobile learning applications.

From the discussion in section 2.5, another research direction is in the further development of Language Design. It has been almost a decade since the latest version of IMS LD and SCORM were published. There are new needs for Language Design due to the recent advances in information technology and research in learning methods. With rapid increase in the use of mobile smart devices, it is important that IMS LD and SCORM provide guidance for the inclusion of resources for mobile platforms such as Android and IOS in addition to conventional web based applications. Specifically, mobile devices have different screen size and capabilities. As a result, IMS LD needs to specify resources in a way which works on devices with different specifications. This research direction will result in more suitable and updated language design which considers new technology, learning resource types, and learning methodologies. Also, the current IMS-LD and SCORM standards are very verbose. More concise standards may encourage the adoption of language design.

Also, for Language Design standards such as IMS LD to flourish, easy-to-use publisher tools to generate learning packages are needed. More needs to be done to improve the usability of the publisher tools as it imposes a high complexity for publishers (Koper, 2005b). Also, personalization in IMS LD compliant learning activities are currently defined during the design stage using IMS LD Level B condition statements by publishers. This places a heavy burden on publishers. Intelligent systems could be designed so that applications can intelligently personalize the content without detailed input from the publishers. Also, to provide incentives for publishers to develop quality Learning packages, a thriving market place, like iTunes for music, App Store for IOS applications is required for Learning activities and resources.

From the discussion from the section 2.6, most of the current research in context adaptation focuses on adaptation of learning resources such as adapting resources to suitable formats based on form-factors e.g., the screen size and internet connection status (Gómez and Fabregat, 2010). There is some work reported which make uses of contextual information for content selection. However, these studies are mostly limited to the use of learner’s location to retrieve learning materials (Chen and Li, 2010; Hsieh, Chen, and Hong, 2007). For example, language learning application uses learner’s location to provide new vocabularies (Chen and Li, 2010) or another application (Chin and Chen, 2013) uses RFID and barcodes to receive additional learning materials. Apart from these, other context information such as social situations and learner cognitive states are not used for content selection. One possible reason could be that these contexts are harder to acquire and quantify. Also, it is important to note that context doesn't necessarily support learning activities e.g., peer support available in the vicinity but could distract learners as well e.g., loud noise or disturbance in the surroundings. The context adaptations should be able to address both aspects as suggested in our examples. Further research is required on how these different types of context can be used in developing context-aware learning applications and how they affect learning activities.

From the discussion in Chapter 5, the functionalities of SCF should be extended by implementing more SCApps. For example, SCF currently does not support the classroom activity recognition. A new SCApp can be developed which gather classroom data from classroom cameras and microphones and analyze the data to infer the current activity of the class. This functionality will be useful for triggering events at appropriate times such as sending classroom notices to students. A new SCApp can be developed to automatically synchronize current classroom learning resources with student mobile devices. As the functionalities and the usefulness of the SCF framework depends on how many SCApps are available in the system, it will be worthwhile to build an open source community around SCF with different educational institutions working together to build up the repository of SCApps.

Moreover, SCHubs currently do not communicate with one another. It would be worthwhile to investigate a way to make multiple SCHubs work together and communicate with one another. It will be useful in a scenario where multiple SCHubs are needed to cover all areas in a large lecture hall or when there are too many students for one SCHub to handle. Also, different SCHubs can then be deployed with different specific tasks. For example, one SCHub can be responsible for student identification process while another can be used for the classroom activity inference and they can exchange data with one another as and when necessary. It would also be interesting to carry out detailed experiments on how the interaction with SCHub affects the battery usage of students’ mobile devices and find ways to optimize it. SCF is still in its first version and there are many directions that we can take from here. However, the current version provides a solid foundation for the future work.

Bibliography

Abet. Retrieved January 6, 2014, from http://www.abet.org.

Abowd, G. D., DEY, A. K., Brown, P. J., Davies, N., Smith, M. and Steggles, P. (1999). Towards a better understanding of context and context-awareness. Handheld and Ubiquitous Computing, Proceedings, 1707, 304-307.

Adl. (2004). SCORM. Retrieved September 12, 2013, from http://www.adlnet.gov/scorm.

Anderson, R., Anderson, R., Davis, K. M., Linnell, N., Prince, C., & Razmov, V. (2007, March). Supporting active learning and example based instruction with classroom technology. In ACM SIGCSE Bulletin (Vol. 39, No. 1, pp. 69-73). ACM.

Antal, M. and Koncz, S. (2011). Student modeling for a web-based self-assessment system. Expert Systems with Applications, 38(6), 6492-6497.

Barron, A. (1998). Designing Web-based Training. British Journal of Educational Technology, 29(4), 355-370.

Bravo, J., Hervas, R., & Chavira, G. (2005). Ubiquitous Computing in the Classroom: An Approach through Identification Process. J. UCS, 11(9), 1494-1504.

Bravo, J., Hervás, R., Nava, S., Chavira, G., Parras, J., Delgado, M. L., ... & Sanz, J. (2007). Towards the Everyday Computing in the Classroom Through Rfid. In Computers and Education (pp. 143-153). Springer Netherlands.

Beck, J., Stern, M., and Haugsjaa, E. (1996). Applications of AI in Education. Crossroads, 3(1), 11-15.

Brusilovsky, P. (1994). The construction and application of student models in intelligent tutoring systems. Journal of Computer and System Sciences International, 32(1), 70-89.

Brusilovsky, P. and Millan, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In The adaptive web (pp. 3-53): Springer-Verlag.

Brusilovsky, P., Schwarz, E. and Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In Intelligent tutoring systems (pp. 261-269): Springer.

Capuano, N., Gaeta, M., Marengo, A., Miranda, S., Orciuoli, F. and Ritrovato, P. (2009). LIA: an intelligent advisor for e-learning. Interactive Learning Environments, 17(3), 221-239.

Capuano, N., Gaeta, M., Micarelli, A. and Sangineto, E. (2002). An integrated architecture for automatic course generation. In Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 02) (pp. 322-326): Citeseer.

Capuano, N., Gaeta, M., Salerno, S. and Mangione, G. R. (2011). An Ontology-Based Approach for Context-Aware E-learning. In Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on (pp. 789-794): IEEE.

Carlson, D. and Schrader, A. (2012). Dynamix: An open plug-and-play context framework for android. In Internet of Things (IOT), 2012 3rd International Conference on the (pp. 151-158): IEEE.

Chen, C.-M., Hsu, S.-H., Li, Y.-L. and Peng, C.-J. (2006). Personalized intelligent m-learning system for supporting effective English learning. In Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on (Vol. 6, pp. 4898-4903): IEEE.

Chen, C. M. and Li, Y. L. (2010). Personalised context-aware ubiquitous learning system for supporting effective English vocabulary learning. Interactive Learning Environments, 18(4), 341-364.

Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. In: Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chin, K. Y. and Chen, Y. L. (2013). A Mobile Learning Support System for Ubiquitous Learning Environments. Proceedings of the 2nd International Conference on Integrated Information (Ic-Ininfo 2012), 73, 14-21.

Clancey, W. J. (1982). Methodology for Building an Intelligent Tutoring System. In: DTIC Document.

Dalziel, J. (2003). Implementing learning design: The learning activity management system (LAMS). In.

Das, M., Bhaskar, M., Chithralekha, T. and Sivasathya, S. (2010). Context Aware E-Learning System with Dynamically Composable Learning Objects. International Journal on Computer Science and Engineering, 2(4), 1245-1253.

De Arriaga, F., Gingell, C., De Arriaga, A., Arriaga J. and Arriaga, F. (2008). A General Student's Model Suitable for Intelligent E-Learning Systems. In Proceedings of the 2nd conference on European computing conference (pp. 167-172): World Scientific and Engineering Academy and Society (WSEAS).

De Jong, T., Specht, M. and Koper, R. (2007). Contextualized media for learning. Educational Technology and Society, 11(2), 41-53.

Derntl, M., Neumann, S. and Oberhuemer, P. (2011). Propelling Standards-based Sharing and Reuse in Instructional Modeling Communities: The Open Graphical Learning Modeler (OpenGLM). In Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on (pp. 431-435): IEEE.

Economides, A. A. (2009). Adaptive context-aware pervasive and ubiquitous learning. International Journal of Technology Enhanced Learning, 1(3), 169-192.

Falquet, G. and Ziswiler, J.-C. (2005). A virtual hyperbooks model to support collaborative learning. International Journal on E-Learning, 4(1), 39-56.

Gomez, S. and Fabregat, R. (2010). Context-aware content adaptation in mlearning. In Proceedings of the 9th World Conference on Mobile and Contextual Learning (pp. 76-83).

Gomez, S., Zervas, P., Sampson, D. G. and Fabregat, R. (2012). Delivering Adaptive and Context-Aware Educational Scenarios via Mobile Devices. Advanced Learning Technologies (ICALT), 197-201.

Goodyear, P. and Retalis, S. (2010). Technology-enhanced learning: Sense Publishers.

Graesser, A. C., Chipman, P., Haynes, B. C. and Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. Ieee Transactions on Education, 48(4), 612-618.

Heeter, C. (1999). Technology Enhanced Learning. Internet 2nd Socio technical Summit.

Horton, W. K. (2000). Designing web-based training: How to teach anyone anything anywhere anytime (Vol. 1): Wiley.

Hsieh, H.-C., Chen, C.-M. and Hong, C.-M. (2007). Context-aware ubiquitous English learning in a campus environment. In Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on (pp. 351-353): IEEE.

Hsu, C.-K., Hwang, G.-J. and Chang, C.-K. (2013). A personalized recommendation-based mobile learning approach to improving the reading performance of EFL students. Computers and Education, 63(0), 327-336.

Hwang, G. J., Tsai, C. C. and Yang, S. J. H. (2008). Criteria, strategies and research issues of context-aware ubiquitous learning. Educational Technology and Society, 11(2), 81-91.

Hwang, G. J., Yang, T. C., Tsai, C. C. and Yang, S. J. H. (2009). A context-aware ubiquitous learning environment for conducting complex science experiments. Computers and Education, 53(2), 402-413.

IMS. (2003). IMS Learning Design Best Practice and Implementation Guide. Retrieved September 12, 2013, from http://www.imsglobal.org/learningdesign.

Kelly, D. and Tangney, B. (2006). Adapting to intelligence profile in an adaptive educational system. Interacting with Computers, 18(3), 385-409.

Koper, R. (2005a). An introduction to Learning design. In Learning design (pp. 3-20): Springer Berlin Heidelberg.

Koper, R. (2005b). Current research in learning design. Educational Technology and Society, 9(1), 13-22.

Kritikou, Y., Demestichas, P., Adamopoulou, E., Demestichas, K., Theologou, M. and Paradia, M. (2008). User Profile Modeling in the context of web-based learning management systems. Journal of Network and Computer Applications, 31(4), 603-627.

Lim, T. S., Sim, S. C., & Mansor, M. M. (2009, October). RFID based attendance system. In Industrial Electronics & Applications, 2009. ISIEA 2009. IEEE Symposium on (Vol. 2, pp. 778-782). IEEE.

Milligan, C. D., Beauvoir, P. and Sharples, P. (2005). The Reload learning design tools. Journal of Interactive Media in Education, 2005(1).

Mor, Y. and Winters, N. (2007). Design approaches in technology-enhanced learning. Interactive Learning Environments, 15(1), 61-75.

NCEES. Retrieved January 6, 2014, from http://ncees.org.

Qu, K. and He, W. (2009). SCORM versus IMS-LD: discussion on development trends of e-learning. In Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on (pp. 1-4): IEEE.

Ruiz, M. D. P., Diaz, M. J. F., Soler, F. O. and Perez, J. R. P. (2008). Adaptation in current e-learning systems. Computer Standards and Interfaces, 30(1-2), 62-70.

Rumetshofer, H. and Wob, W. (2003). Individualized e-learning systems enabled by a semantically determined adaptation of learning fragments. In Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on (pp. 640-645): IEEE.

Sampson, D., Karagiannidis, C. and Kinshuk, D. (2010). Personalised learning: educational, technological and standarisation perspective. Digital Education Review (4), 24-39.

Sampson, D. G., Isaias, P., Ifenthaler, D. and Spector, J. M. (2012). Ubiquitous and mobile learning in the digital age: Springer.

Sanjuan-Martinez, O., G-Bustelo, B. C. P., Crespo, R. G. and Franco, E. T. (2009). Using Recommendation System for E-learning Environments at degree level. International Journal of Interactive Multimedia and Artificial Intelligence, 1(2).

Schiaffino, S., Amandi, A., Gasparini, I. and Pimenta, M. S. (2008). Personalization in e-learning: the adaptive system vs. the intelligent agent approaches. In Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems (pp. 186-195): Sociedade Brasileira de Computação.

Schilit, B., Adams, N. and Want, R. (1994). Context-aware computing applications. In Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on (pp. 85-90): IEEE.

Shishehchi, S., Banihashem, S. Y. and Zin, N. A. M. (2010). A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system. In Information Technology (ITSim), 2010 International Symposium in (Vol. 1, pp. 1-5): IEEE.

Shishehchi, S., Banihashem, S. Y., Zin, N. A. M., Noah, S. A. M. and Malaysia, K. (2012). Ontological approach in knowledge based recommender system to develop the quality of e-learning system. Australian Journal of Basic and Applied Sciences, 6(2), 115-123.

Schneider, P. (2010). Engaging Students by Efficiently Monitoring Attendance and Participation. In 13th Pacific Rim First Year in Higher Education (FYHE) Conference.

Shi, Y., Xie, W., Xu, G., Shi, R., Chen, E., Mao, Y., & Liu, F. (2003). The smart classroom: merging technologies for seamless tele-education. IEEE Pervasive Computing, 2(2), 47-55.

Shute, V. J. and Zapata-Rivera, D. (2012). Adaptive Educational Systems. Adaptive Technologies for Training and Education, 7.

Siadaty, M., Torniai, C., Gasevic, D., Jovanovic, J., E., T. and Hatala, M. (2008). m-LOCO: An ontology-based framework for context-aware mobile learning. Proceedings of the 6th International Workshop on Ontologies and Semantic Web for Intelligent Educational Systems collocated with the 9th International Conference on Intelligent Tutoring Systems.

Sudhana, K., Raj, V. C. and Suresh, R. (2013). An ontology-based framework for context-aware adaptive e-learning system. In Computer Communication and Informatics (ICCCI), 2013 International Conference on (pp. 1-6): IEEE.

Suo, Y., Miyata, N., Morikawa, H., Ishida, T., & Shi, Y. (2009). Open smart classroom: Extensible and scalable learning system in smart space using web service technology. Knowledge and Data Engineering, IEEE Transactions on, 21(6), 814-828.

Tankeleviciene, L. and Damasevicius, R. (2009). Towards a conceptual model of learning context in e-learning. In Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on (pp. 645-646): IEEE.

Taxila, P. (2009). Development of Academic Attendence Monitoring System Using Fingerprint Identification. IJCSNS, 9(5), 164.

Thus, H., Chatti, M. A., Yalcin, E., Pallasch, C., Kyryliuk, B., Mageramov, T. and Schroeder, U. (2012). Mobile learning in context. International Journal of Technology Enhanced Learning, 4(5), 332-344.

Triantafillou, E., Pomportsis, A. and Georgiadou, E. (2002). AES-CS: adaptive educational system based on cognitive styles. Adaptive Systems for Web-Based Education, 10-20.

Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P. and Koper, R. (2005). Authoring a full life cycle model in standards-based, adaptive e-learning.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I. and Duval, E. (2012). Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. IEEE Transactions on Learning Technologies, 5(4), 318-335.

Viet Anh, N., Van Cong, P. and Si Dam, H. (2010). A Context - Aware Mobile Learning Adaptive System for Supporting Foreigner Learning English. In Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on (pp. 1-6).

Yu, Z., Nakamura, Y., Jang, S., Kajita, S. and Mase, K. (2007). Ontology-based semantic recommendation for context-aware e-learning. In Ubiquitous Intelligence and Computing (pp. 898-907): Springer.

Zapata-rivera, J.-D. and Greer, J. (2001). Externalising learner modelling representations. In Proceedings of Workshop on External Representations of AIED: Multiple Forms and Multiple Roles (pp. 71-76).

Zervas, P., Ardila, S. E. G., Fabregat, R. and Sampson, D. G. (2011). Tools for context-aware learning design and mobile delivery. In Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on (pp. 534-535): IEEE.

Zhu, J., Chen, P., Pung, H., Oliya, M., Sen, S. and Wong, W. C. (2011). Coalition: A Platform for Context-Aware Mobile Application Development. UbiCC Journal, 722-735.