Ontology-Based Recommendation Algorithms for Personalized Education

Amir Bahmani[1], Sahra Sedigh[2], and Ali Hurson[1]

[1] Department of Computer Science,
[2] Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
{abww4,sedighs,hurson}@mst.edu

Abstract. This paper presents recommendation algorithms that personalize course and curriculum content for individual students, within the broader scope of Pervasive Cyberinfrastructure for Personalizing Learning and Instructional Support (PERCEPOLIS). The context considered in making recommendations includes the academic background, interests, and computing environment of the student, as well as past recommendations made to students with similar profiles. Context provision, interpretation, and management are the services that facilitate consideration of this information. Context modeling is through a two-level hierarchy of generic and domain ontologies, respectively; reducing the reasoning search space. Imprecise query support increases the flexibility of the recommendation engine, by allowing interpretation of context provided in terms equivalent, but not necessarily identical to database access terms of the system. The relevance of the recommendations is increased by using both individual and collaborative filtering. Correct operation of the algorithms has been verified through prototyping.

1 Introduction

The growing ubiquity of mobile devices and the rapid growth of wireless networks have facilitated “anytime, anywhere” access to information repositories [11]. Moreover, the increasingly seamless integration of computing devices with daily life has brought about the “pervasive computing” paradigm - the key technology enabling this integration. By the same token, “pervasive learning,” where transparent computing is leveraged to enhance education, has emerged as an application [10].

One of the critical functionalities of pervasive computing environments in general, and pervasive learning in particular, is service discovery. In pervasive environments, service discovery mechanisms must overcome intrinsic hardware, software, and networking heterogeneity [16] and support learners with a sufficient selection of services that meet their current requirements; i.e., context awareness. Applications in pervasive and ubiquitous computing utilize context to capture differences and adapt the application's behaviors to the operating environment [15]. Context-aware computing is defined in Ref. [6] as the use of context in software applications that adapt their behavior based on the discovered contexts.

In our work, knowledge of the operating context (including the profile of the operator) is used to personalize courses and curricula for individual students, based on their academic profile, interests, and learning style. This paper describes the methodology that brings context-awareness to Pervasive Cyberinfrastructure for Personalized Learning and Instructional Support (PERCEPOLIS) to create an adaptive learning environment that facilitates resource sharing, collaboration, and personalized learning. PERCEPOLIS utilizes pervasive computing through the application of intelligent software agents to facilitate modular course development and offering, blended learning, and support of networked curricula [3].

Modularity, blended learning, and support of networked curricula are three essential attributes of PERCEPOLIS. Modularity increases the resolution of the curriculum and allows for finer-grained personalization of learning artifacts and associated data collection. Blended learning allows class time to be used for active learning, interactive problem-solving, and reflective instructional tasks, rather than for traditional lectures. In networked curricula, different courses form a cohesive and interconnected whole, and learning in one area reinforces and supports learning in others.

Fig. 1 depicts the personalization of a computer architecture course (CS 388) within the CS curriculum. Beginning from the top of the figure, modularity can be seen in successively decomposing the curriculum into courses, subject topics, and finally modules. The connections between these entities illustrate the networked curriculum. For each module, alternatives can be developed that differ from each other in level of difficulty, use of multimedia, appropriateness for out-of-class study (in blended learning), and other features. PERCEPOLIS recommends one of these alternatives, based on context that includes the student profile, his or her access environment, and the content of each module.

Pervasive learning overcomes the limitations of traditional passive, lecture-based classroom platform by allowing a student to peruse learning resources outside of class. Personalization strives to ensure that the learning resources recommended to each student are the most appropriate for him or her. To determine a personalized course trajectory for each student, PERCEPOLIS requires a complex recommender system that leverages computational intelligence to recommend learning artifacts and resources; such as books, hyperlinks, courses, and modules based on each student's profile and recommendations made to students with similar profiles [3].

The remainder of this paper is organized as follows. Section 2 provides a brief survey of related research as advanced in the literature. The major components of the proposed context-aware system are introduced in Sect. 3. The workflow and prototype of the proposed model are given in Sections 4 and 5, respectively. The paper closes with concluding remarks that includes avenues considered for extensions to this research.

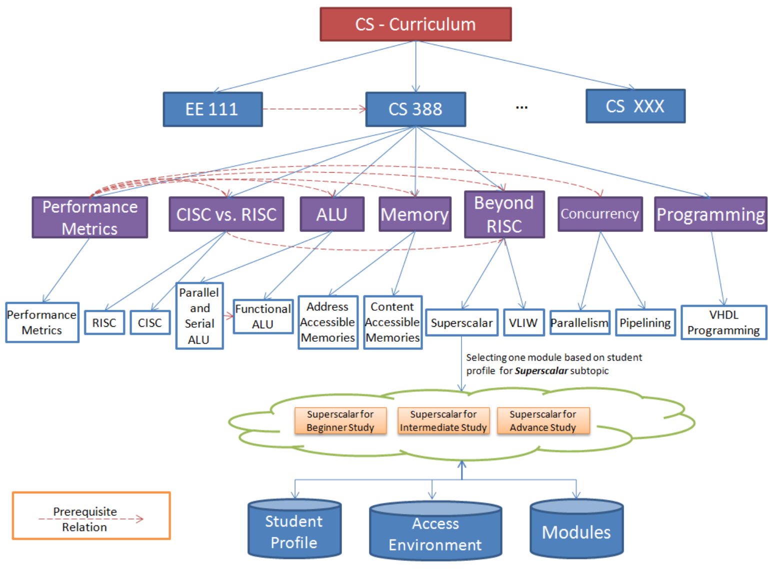

Fig. 1. Personalization of a computer architecture course (CS 388)

2 Related Literature

Context-awareness is a core attribute of PERCEPOLIS, and in large part responsible for the intelligence of the system. The underlying data structures play a significant role in the representation and exchange of context information. Key-value pair, markup-scheme, graphical, object-oriented, logic-based, and ontology-based models are data structures whose use for context representation is described and evaluated in [14]. Ontology-based models are cited as showing the greatest utility in pervasive computing, due their formal expressiveness and the fact that they facilitate reasoning [4]

Ontologies provide a controlled vocabulary of concepts, each with explicitly-defined and machine-processable semantics. By defining shared and common domain theories, ontologies allow humans and machines to communicate concisely, supporting exchange of not only syntax, but also semantics. Inherent to the use of ontologies is a trade-off between expressiveness and complexity of reasoning [5,8]. Abundant literature exists on context-aware frameworks that utilize ontologies as the underlying context model [11], [8] and [2].

The C-CAST context management architecture was proposed in [11]. It supports mobile context-based services by decoupling provisioning and consumption. The decoupling is considered in the model to enhance scalability in terms of physical distribution and context reasoning complexity. The system is built based on three basic functional entities: the context consumer (CxC), context broker (CxB), and context provider (CxP).

A context-aware framework (CAF) developed for the HYDRA middleware project is described in [2]. The intent of HYDRA was to develop middleware to support an intelligent networked embedded system based on a service-oriented architecture. The CAF in HYDRA underpins the context-aware applications and services, while being domain-agnostic and adaptable. The CAF contains two core components: the data acquisition component and the context manager.

The novelty of our approach is threefold. Firstly, the suggested context-aware system captures and integrates capabilities of existing models. Secondly, it utilizes the summary schema model (SSM) to overcome heterogeneity and provide transparency. Thirdly, the proposed recommendation algorithm not only leverages both content-based and collaborative filtering techniques, but it can also consider a student's physical capability and adapt the content of recommended objects accordingly. The proposed system is discussed in detail in Sect. 3.

2.1 Intelligent Software Agents

A *software agent* is a computer program that acts autonomously on behalf of a person or organization. Agents can be particularly beneficial in pervasive learning environments, as they can assist in transparently managing information overload [13]. Leveraging pervasive computing and communications at various levels through the use of agent-based middleware is a defining feature of PERCEPOLIS. A number of existing personalized learning systems, enumerated below and reported in [9] and [10], similarly employ multi-agent systems.

ISABEL [9] uses four types of agents: 1) device agents that monitor and profile each student's access device, 2) student agents that construct a complete profile of each student's interests, 3) tutor agents that interact with and identify similarities among a group of student agents characterized by a specific domain of interest, and 4) teacher agents that are associated with and manage the learning artifacts of an e-learning suite.

The pervasive learning infrastructure reported in Ref. [10] uses four types of agents. Location-aware learner agents are created for each learner logged in within a specific coverage area. Each learner agent uses the learner's preferences or previous behavior to populate the student model used by the infrastructure for storing and updating relevant information about learners. Connection agents are responsible for managing the connection between the mobile devices and the agent platform. Service agents are available for each service provided by the infrastructure. Finally, resource agents are responsible for managing resources.

PERCEPOLIS uses agents similarly. As articulated in Sect. 4, an agent is created for each instructor/advisor, course, and student; respectively. These agents communicate with each other to determine a recommendation.

3 Context-Awareness in PERCEPOLIS

In this section, we propose a context-aware system that introduces new features while maintaining the benefits of existing context-aware frameworks. As depicted

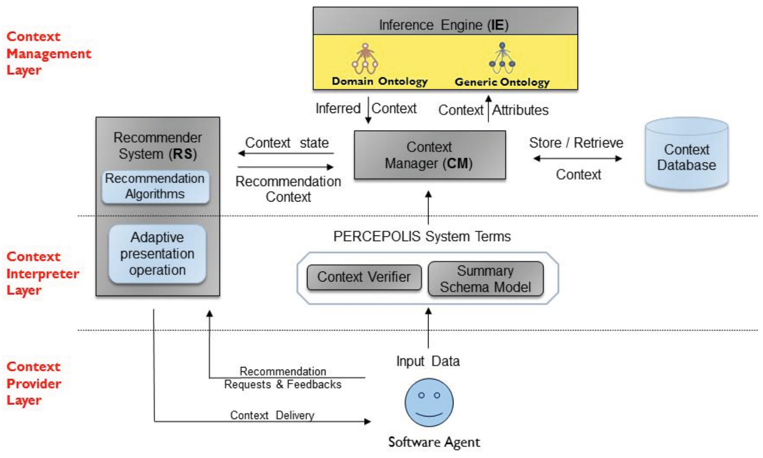

Fig. 2. Context-awareness in PERCEPOLIS

in Fig. 2, the proposed context-aware system includes three main layers: 1) the context provider layer, 2) the context interpreter layer, and 3) the context management layer.

Two types of contextual information - explicit and inferred - are captured in these layers and utilized in PERCEPOLIS. *Explicit contextual information* is provided directly by the learner or institution by completing surveys. This information can be classified into five categories:

1. Learner profile: Includes academic records (list of courses and modules passed, grades, GPA, target degree, major, etc.) and personal profile (location, disabilities, interests, needs and skills).
2. Course profile: Includes information such as department ID, mandatory topics elective topics, and default modules (identified as core content by the instructor of a course).
3. Topic profile: Describes and enumerates attributes of a topic.
4. Subtopic profile: Describes and enumerates attributes of a subtopic.
5. Module profile: Includes information such as prerequisites, contents (by topic and learning artifact), nominal level of difficulty (beginner, intermediate or advanced), and developer of module.
6. Instructor profile: Includes courses taught, skills, research interests, etc.

Inferred contextual information falls into one of two categories:

1. Learner tacit profile: Includes learning style; the learner's infrastructure (device, operating system, networking); access records; tacit skills, e.g., passing a certain module may enable a new skill; skill level; tacit interests, e.g., passing a certain module with a high grade may reflect the learner's interest in that topic.

2. Module tacit profile: Includes the implicit level of difficulty (inferred from grades) and the audience (based on frequency of use in specific courses or learners who have taken the module).

3.1 Context Provider Layer

Student and instructor agents in this layer are responsible for capturing contexts and forwarding the captured information to the layer above - the context interpreter layer. PERCEPOLIS software agents and their tasks are described in Sect. 4.

3.2 Context Interpreter Layer

Heterogeneity, which complicates transparency and scalability of computing systems, is an intrinsic characteristic of pervasive environments [16]. In PERCEPOLIS, two functions of the context interpreter layer address these challenges: 1) context verification and 2) adaptive presentation.

One of the main impediments to transparency is the difference between terminology used by developers and users, respectively. As an example, in describing a topic, the system or module developer may have used the term "*arithmetics*," but the user query may specify "*mathematics*." Unless translation takes place, a null response will be returned to the user - despite the existence of learning artifacts relevant to mathematics. In our system, the context verifier utilizes the *summary schema model* (SSM) for reconciling terminology. The SSM creates a taxonomy/ontology based on the linguistic meaning and relationships among terms.

3.3 Context Management Layer

The information resulting from context interpretation is forwarded to the context management layer, which houses the PERCEPOLIS ontology and recommendation algorithms. In the interest of scalability and efficiency, we utilize a two-tier scheme that includes generic and domain ontologies, respectively.

The filtering techniques used by recommender systems fall into one of two categories - *content-based* or *collaborative* filtering [15]. Content-based filtering techniques focus solely on identifying resources based on the profile of an individual or artifact. In contrast, collaborative filtering techniques take a peer group approach - recommendations are made based on similarities among individuals or artifacts. [3].

To prioritize and recommend the most appropriate learning artifacts; e.g., courses or topics; the recommender system must find associations among the learning artifacts and the preferences, skills, and background of a student. Ontologies can be used to represent and facilitate later identification of such relationships. We utilize the ACM Computing Classification System [1] as the common ontology in PERCEPOLIS. Fig. 3 includes a portion of the ACM CCS that decomposes hardware into six constituent topics.

To place increased emphasis on the relationships between modules and student preferences, we defined two parameters: *Relevancy Weight* (RW) and *Preference Weight* (PW). RW quantifies the relevance of a module to particular topic; PW quantifies the interest of a student in a particular topic. For instance, in Fig. 3, the RW between the module "*ALU for intermediate study*" and the topic "*Arithmetic and Logic Structures*" is 0.3. The PW between the same topic and the student in question is 0.6.

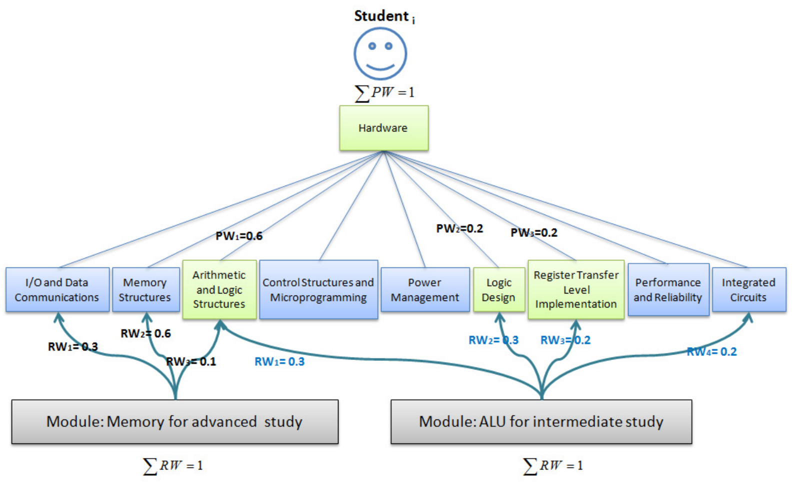
Student $_i$
$\sum PW = 1$
Hardware
PW₁=0.6
PW₂=0.2
PW₃=0.2
I/O and Data Communications
Memory Structures
Arithmetic and Logic Structures
Control Structures and Microprogramming
Power Management
Logic Design
Register Transfer Level Implementation
Performance and Reliability
Integrated Circuits
RW₁= 0.3
RW₂= 0.6
RW₃= 0.1
RW₁= 0.3
RW₂= 0.3
RW₃= 0.2
RW₄= 0.2
Module: Memory for advanced study
Module: ALU for intermediate study
$\sum RW = 1$
$\sum RW = 1$

Fig. 3. Relevancy and preference weights

One task of context management is to create an individual preference tree (IPT) for each student. IPTs are created based on domain ontology and explicit and implicit contextual information. In Fig. 3, the arcs labeled with "PW" values and the nodes they connect represent the IPT of student$_i$.

Both content-based and collaborative filtering require quantification of the similarity of one or more entities. To this end, we utilize Pearson correlation, which quantifies the extent to which two variables linearly relate with each other [12], [7].

4 Workflow of the Recommendation Scheme

PERCEPOLIS recognizes three sets of entities as comprising the educational environment: i) the set of instructors/advisors, I; ii) the set of students, S; and iii) the set of courses, C. Each course $c \in C$ is a collection of interrelated mandatory and elective modules. Each of I, S, and C, respectively; is represented by a community of software agents that communicate and negotiate with each other to determine the best trajectory for each student through a course or curriculum.

More specifically, each student has an agent, S$_A$, which is responsible for acquiring from the context provider layer information related to a student's profile; e.g., student ID, name, department, and explicit interests/skills. S_A is subsequently responsible for sending this information to the context interpreter layer. With the help of the context verifier and SSM, the context interpreter layer interprets the input data, reconciling terminology when needed. The resulting context information is sent to the context management layer. As mentioned previously, the use of the generic ontology results minimizes the reasoning search space. For instance, if a student needs a home department course, instead of searching among all courses offered by the university, the search space is reduced to the student's department. The Inference Engine (IE) will check the provided context information against the generic ontology; finally all captured context information will be stored in Context Database.

S_A passes recommendation requests (made by a user through the user interface) to the recommender system (RS), which in turn identifies the appropriate artifacts. The personalization process depicted in Fig. 4 begins after the RS receives a recommendation request. The RS contacts the Context Manager (CM) to request the information considered in personalization, which includes IPTs, department rules, a list of courses, and course relevancy weights. IPTs are created by the CM based on student preferences and topic and module relevancy weights that are assigned by module developers.

Finally, after the student selects the desired subtopics, the RS identifies the most appropriate modules, based on student's background and interests and other context information. The content of the modules will be adapted based on the functionality and capability of the student's device using adaptive presentation . For instance, if a student's device is the Samsung SGH A107, then the IE will infer that the cell phone does not support video services, and hence will avoid recommendation of artifacts that include video.

5 PERCEPOLIS Prototype

As of publication of this paper, we have completed the first PERCEPOLIS prototype - including the proposed context-aware system. The prototype and profile databases have been implemented in Java SE 6 and MySQL 5.5.8, respectively. The prototype allows for operation in one of two modes: 1) *administrator*, and 2) *student*. Seven entities can be managed (searched/added/updated/deleted) in administrator mode: 1) words, 2) modules, 3) subtopics 4) topics, 5) courses, 6) SSM, and 7) ontology.

In student mode the system can carry out the following tasks: 1) Display the *Most Recent History*: where students can track the modules recommended by the system, as well as their own final selections; 2) *Update Profile*: where students can provide their interests as explicit context; 3) *Find Appropriate Courses*: where the system recommends the most appropriate courses based on students' academic profiles and departmental rules; and 4) Find appropriate modules: where, after students select their desired courses from among acceptable/recommended courses, the system helps students find the best trajectory of topics/subtopics/modules for each course and allows the students to select their desired subtopics from among acceptable, elective subtopics.

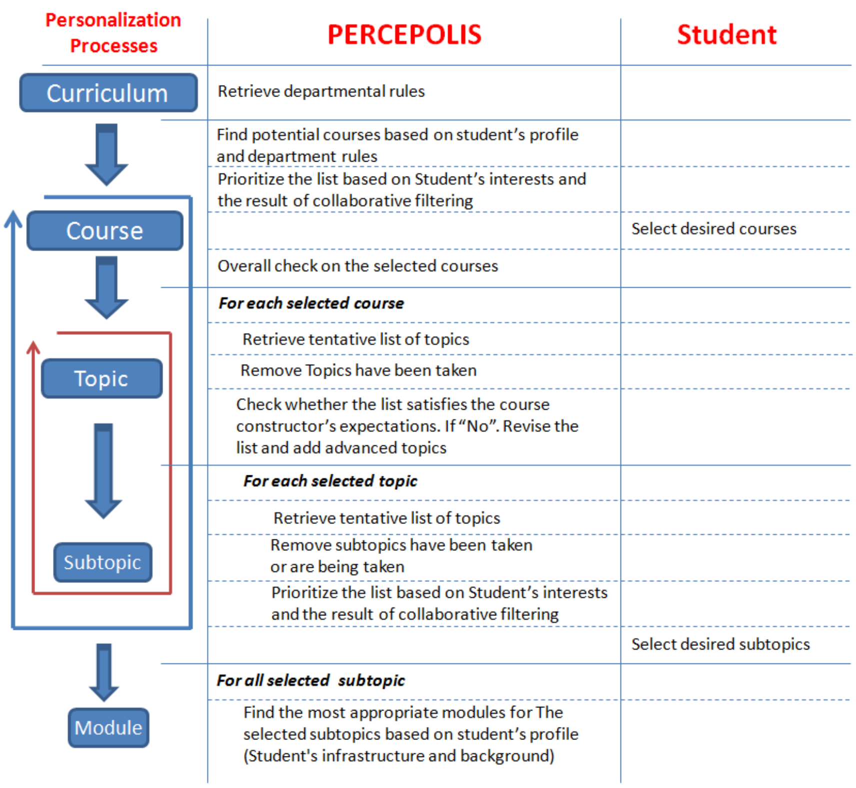
Personalization Processes
PERCEPOLIS
Student
Curriculum
Retrieve departmental rules
Find potential courses based on student's profile and department rules
Prioritize the list based on Student's interests and the result of collaborative filtering
Course
Select desired courses
Overall check on the selected courses
For each selected course
Retrieve tentative list of topics
Topic
Remove Topics have been taken
Check whether the list satisfies the course constructor's expectations. If "No". Revise the list and add advanced topics
For each selected topic
Retrieve tentative list of topics
Subtopic
Remove subtopics have been taken or are being taken
Prioritize the list based on Student's interests and the result of collaborative filtering
Select desired subtopics
For all selected subtopic
Module
Find the most appropriate modules for The selected subtopics based on student's profile (Student's infrastructure and background)

Fig. 4. Personalization process in PERCEPOLIS

6 Conclusion

In this paper, within the scope of PERCEPOLIS, we have presented a new layered context-aware system and its functionalities. The proposed model is composed of three layers: 1) the context provider layer, 2) the context interpreter layer, and 3) the context management layer. The novelty of our model is threefold. Firstly, the suggested context-aware system captures and integrates capabilities of existing models. Secondly, it utilizes SSM to overcome heterogeneity and provide transparency. Thirdly, the proposed recommendation algorithm leverages both content-based and collaborative filtering techniques and adapts the content of recommended artifacts based on a student's physical capability.

References

1. Association for Computing Machinery. The ACM Computing Classification System (1998 version), http://www.acm.org/about/class/1998 (retrieved June 2012)
2. Badii, A., Crouch, M., Lallah, C.: A context-awareness framework for intelligent networked embedded systems. In: Proceedings of the International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services, pp. 105–110 (2010)

3. Bahmani, A., Sedigh, S., Hurson, A.R.: Context-aware recommendation algorithms for the percepolis personalized education platform. In: Proceedings of the Frontiers in Education Conference (FIE), pp. F4E–1–F4E–6 (2011)
4. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2, 263–277 (2007)
5. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing 6(2), 161–180 (2010)
6. Coppola, P., Della Mea, V., Gaspero, L., Lomuscio, R., Mischis, D., Mizzaro, S., Nazzi, E., Scagnetto, I., Vassena, L.: AI techniques in a context-aware ubiquitous environment. In: Hassanien, A.-E., Abawajy, J.H., Abraham, A., Hagras, H. (eds.) Pervasive Computing. Computer Communications and Networks, pp. 157–180. Springer, London (2010)
7. Dowdy, S., Wearden, S., Chilko, D.M.: Statistics for Research. John Wiley and Sons (2004)
8. Ejigu, D., Scuturici, M., Brunie, L.: An ontology-based approach to context modeling and reasoning in pervasive computing. In: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 14–19 (2007)
9. Garruzzo, S., Rosaci, D., Sarne, G.: Isabel: A multi agent e-learning system that supports multiple devices. In: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 485–488 (2007)
10. Graf, S., MacCallum, K., Liu, T., Chang, M., Wen, D., Tan, Q., Dron, J., Lin, F., Chen, N., McGreal, R., Kinshuk, N.: An infrastructure for developing pervasive learning environments. In: Proceedings of the IEEE International Conference on Pervasive Computing and Communications, pp. 389–394 (2008)
11. Knappmeyer, M., Baker, N., Liaquat, S., Tönjes, R.: A Context Provisioning Framework to Support Pervasive and Ubiquitous Applications. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 93–106. Springer, Heidelberg (2009)
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186 (1994)
13. Sakthiyavathi, K., Palanivel, K.: A generic architecture for agent based e-learning system. In: Proceedings of the International Conference on Intelligent Agent Multi-Agent Systems, pp. 1–5 (2009)
14. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Proceeings of the Workshop on Advanced Context Modelling, Reasoning and Management - The Sixth International Conference on Ubiquitous Computing (2004)
15. Xu, C., Cheung, S.C., Chan, W.K., Ye, C.: Partial constraint checking for context consistency in pervasive computing. ACM Transactions on Software Engineering and Methodology 19, 9:1–9:61 (2010)
16. Xu, K., Zhu, M., Zhang, D., Gu, T.: Context-aware content filtering & presentation for pervasive & mobile information systems. In: Proceedings of the 1st International Conference on Ambient Media and Systems, pp. 20:1–20:8 (2008)