Context-Aware Recommendation Algorithms for the PERCEPOLIS Personalized Education Platform

Amir Bahmani, Sahra Sedigh, and Ali R. Hurson
Missouri University of Science and Technology, abww4@mst.edu, sedighs@mst.edu, hurson@mst.edu

Abstract - This paper describes Pervasive Cyberinfrastructure for Personalized Learning and Instructional Support (PERCEPOLIS), where context-aware recommendation algorithms facilitate personalized learning and instruction. Fundamental to PERCEPOLIS are (a) modular course development and offering, which increase the resolution of the curriculum and allow for finer-grained personalization of learning artifacts and associated data collection; (b) blended learning, which allows class time to be used for active learning, interactive problem solving and reflective instructional tasks; and (c) networked curricula, in which the components form a cohesive and strongly interconnected whole where learning in one area reinforces and supports learning in other areas. Intelligent software agents customize the content of a course for each learner, based on his or her academic profile and interests, aided by context-based recommendation algorithms. This paper provides an introduction to the PERCEPOLIS platform, with focus on these algorithms; and describes the educational research that underpins its design.

Index Terms – Context-aware recommendation, Multi-agent software, Personalized learning, Pervasive computing.

INTRODUCTION

A *pervasive learning* environment is defined as a setting where learners can become completely immersed in the learning process [1]. The key technology enabling such learning environments is *pervasive computing*, where a wide variety of computing devices are transparently and gracefully leveraged for enrichment of our living and working spaces [2]. Advances in databases, distributed computing, computational intelligence, and especially pervasive computing can be used to fundamentally transform higher education and instructional design [3]. The pervasive learning facilitated by these technologies overcomes the limitations of traditional passive lecture-based classroom learning by providing learning materials to learners according to their profile, which includes information such as learning style, interests, level of knowledge, and goals. These abilities result from the anytime, anywhere access to educational materials facilitated by pervasive computing, and the adaptive and personalized learning that results from dynamic and intelligent recommendation of learning artifacts to each learner [1, 4, 5].

Critical to the efficacy of this personalized learning is context-awareness of the recommendation procedure, which necessitates that context information be extracted, interpreted, and utilized for personalization by the underlying cyberinfrastructure. Context-awareness further requires that the functionality of the pervasive learning system be adapted based on its context at the time of use, leveraging context to supply applicable information and/or services to the user based on his or her behavior [6, 7]. More specifically, pervasive learning environments provide context-aware discovery and acquisition of the most appropriate educational resources from a potentially massive base [8].

This paper describes *Pervasive Cyberinfrastructure for Personalized Learning and Instructional Support (PERCEPOLIS)*, which leverages context-aware pervasive computing to create an adaptive learning environment that facilitates resource sharing, collaboration, and personalized learning in higher education [3]. PERCEPOLIS promotes and enables three key changes to the currently dominant pedagogy: modular course development and offering, blended learning, and networked curricula. *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 traditional lectures. In *networked curricula*, different courses form a cohesive and interconnected whole, and learning in one area reinforces and supports learning in others.

Within the scope of pervasive computing, a simple example is a system where a person's cellular phone automatically contacts his or her refrigerator, which responds with a list of its contents to inform the person of whether he or she has a sufficient supply of a particular item. The binary decision required in this example; i.e., whether or not a purchase is necessary, requires only trivial computational intelligence. The decision support required for personalized learning is significantly more sophisticated. In order to determine a personalized course trajectory for each learner, the system must select from a potentially large set the most appropriate learning materials for each learner, based on his or her background, interests, and needs. PERCEPOLIS requires a complex recommender system - as do most other pervasive learning environments, which leverage computational intelligence to recommend materials/resources; e.g., books, hyperlinks, and courses,

based on each learner's profile and recommendations made to learners with similar profiles [8, 9].

As a result of inadequate filtering techniques, the recommender systems of existing pervasive learning platforms effectively ignore the dynamic interests and preferences, access patterns, and other attributes of learners [8]. One goal of PERCEPOLIS is to remedy this shortcoming, using a context-aware resource recommendation model.

A noteworthy aspect of usage neglected by existing pervasive learning systems is the relevance of specific attributes, in particular environmental attributes, under given conditions. As an example, the networking capabilities of the user's end system should play a significant role in determining the educational artifacts to be recommended, but are frequently neglected due to focus of the recommender system on matching the contents of artifacts with attributes directly related to the learner [8]. In contrast, the recommender system of PERCEPOLIS takes into account the attributes of both the learner and his or her environment.

In brief, the novelty of PERCEPOLIS lies in its ability to leverage pervasive computing and communication through the use of intelligent software agents that use a learner's academic profile and interests, as well as supplemental information such as his or her learning style and environment, to customize the content of a course for the learner [3]. Moreover, PERCEPOLIS is a global information sharing platform that acts as middleware connecting: a) databases housing learner profile information and b) instructional platforms or databases where educational artifacts are hosted. Figure 1 depicts an overview of the cyberinfrastructure.
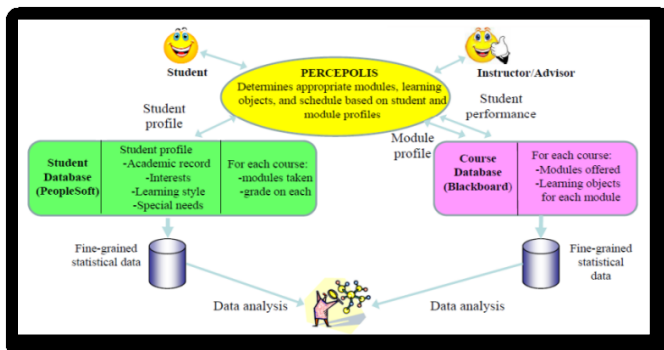

FIGURE 1
OVERVIEW OF PROPOSED CYBERINFRASTRUCTURE

The remainder of this paper is organized as follows. In the next section, we provide a brief survey of related literature. The major components of PERCEPOLIS are subsequently introduced. The final section concludes the paper and describes enhancements planned for the platform.

RELATED RESEARCH

The computational intelligence that facilitates personalized learning in PERCEPOLIS relies on two major technologies: intelligent software agents and a context-aware recommender system [3]. A number of studies related to each category are summarized in this section of the paper.

I. Intelligent Software Agents

A *software agent* is a computer program that acts autonomously on behalf of a person or organization [10]. Agents can be particularly beneficial in pervasive learning environments, as they can assist in transparently managing information overload [11]. 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 similarly employ multi-agent systems. We enumerate them below.

Information Software Agent-Based E-Learning (ISABEL) is a platform that enables interaction between users and e-learning web sites and provides helpful suggestions about the educational resources available to learners [12]. ISABEL 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 site.

A pervasive learning infrastructure based on multi-agent system architecture is proposed in [13]. The infrastructure uses four types of agents. *Location-aware learner agents* are created for each learner logged in within a specific coverage area. An 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.

An adaptable and intelligent architecture for web-based distance education is presented in [14]. Three types of agents are utilized: 1) *assistant agents* that interact with students, 2) *evaluation agents* that update the student profile after evaluating the student, 3) *pedagogical agents* that generate and update course content based on student preferences, and 4) *expert agents* that solve problems and exercises are pertinent to a course.

The design and development of an integration platform that enhances assessment agents for e-learning environments is presented in [15]. The proposed agent platform can support various intelligent agents that provide assessment services based on computational intelligence techniques such as Bayesian networks and genetic algorithms.

An Agent Based Intelligent Tutoring System for Distance Learning (ABITS) is proposed in [16]. Three types of agents are employed in the system: 1) *evaluation agents* responsible for evaluating and updating student models; i.e., the cognitive state and learning preferences for each student; 2) *pedagogical agents* that evaluate and update curricula; and 3) *affective agents* that evaluate and update learning preferences.

In Section 3, we discuss the intelligent software agent model employed by PERCEPOLIS. We utilize mobile agents for more dynamic and robust implementation of the artifacts developed. Resulting advantages include reduced communication cost and power consumption and improved response time, as reported in [17]. We enable inference capability in the agents by providing special services in the proposed context-aware recommender system, as subsequently discussed. This intelligence reduces the communication cost between learners and the environment - a key concern in pervasive systems [18].

II. Context-Based Recommender Systems

In this section, we first define context-aware recommender systems; then discuss related literature; and finally, describe the capabilities of our context-aware recommender system.

Recommender systems assist users in making an informed selection of one or more items; e.g., books, articles, movies; from a pool of candidates [19]. The *context* considered by such systems in making the recommendation is broadly defined as any information that can be used to characterize an entity such as a person [20].

The educational recommender systems developed over the past decade have considered only two types of entities: learners and items, and do not consider context information in making recommendations. However, context-aware resource recommendation can play an important role in pervasive learning environments [8]. Two general approaches to leveraging contextual information in recommendation processes are (1) recommendation via context-driven query and search, and (2) recommendation via contextual preference elicitation and estimation [21].

In the first approach, the obtained contextual information is used to submit a query or search a repository of resources, and then present the most appropriate matching resources to the learner. In contrast, the second approach tries to understand and model the needs and interests of each learner by following his or her interactions (as well as those of other learners) with the educational system, or by receiving preference feedback from the learner on previously recommended learning artifacts.

The influence of pervasive games on English learning achievement and motivation is investigated in [22], through a context-aware pervasive learning environment denoted as Handheld English Language Learning Organization (HELLO). The system utilizes sensors, augmented reality, the Internet, pervasive computing, and related information technologies.

JAPELAS is a context-aware support system for the learning of formal expressions in Japanese [23]. The system can recommend appropriate expressions to learners, according to his or her situation and personal profile.

ePH, a system that enables the sharing of public-interest information that can be accessed via always-on, context-aware services has been described in [24]. A multi-agent architecture and multi-dimensional context model are employed by the system.

Addressing the gap between the learning accomplished during indoor computer-based learning activities in comparison to outdoor field trips is the objective of the system described in [25]. The solution proposed is the use of pervasive learning systems where mobile devices can be used to collect and report contextual information, which can be commented on by other users who may be in different physical or virtual environments.

PERKAM is a pervasive computing environment that allows learners to share knowledge, interact, collaborate, and exchange individual experiences. Radio-frequency identification (RFID) is used to identify and profile the learner, objects, location, and environment; and to recommend the most appropriate learning materials [26].

The following four drawbacks have been enumerated in [8] for the learning context-based recommender systems currently used in pervasive learning systems.

- Existing recommendation algorithms are based on either content-filtering or collaborative recommendation algorithms. The authors assert that neither category is sufficient on its own.

- The recommendation techniques surveyed do not take into account the access time of historical records. If the learners' interests change with the lapse of time, this change will not be observed.

- Due to the repeatability and periodicity of the learning process, dependence relationships are likely to occur among learners' historical access records. These relationships are not taken into account by the recommender systems.

- These systems focus on logical attributes; e.g., similarity among learners' preferences, and neglect situational attributes. For instance, pervasive learning environments should support a broad range of devices, from desktop computers to smart phones. Consequently, they should be able to account for device (and network) capabilities when recommending learning artifacts.

We describe the context-based recommender system employed by PERCEPOLIS in the following section. We employ both content-based filtering and collaborative filtering techniques and account for the dynamic nature of learning processes and environments.

FEATURES AND COMPONENTS OF PERCEPOLIS

One of the key features of PERCEPOLIS is its modular approach to course development and offering, which enables finer-grained personalization of learning and data collection processes by increasing the resolution of the curriculum.

Each course is decomposed into several content modules - some mandatory, and others that are elective. Mandatory modules are dictated by course and curriculum objectives, and elective modules can be chosen to supplement the learner's knowledge of prerequisites or to engage an interested learner in more advanced topics. Each module as a standalone object has its own learning artifacts, such as prerequisite modules, lecture notes, problems, sample solutions, and programming or laboratory exercises. Modules in different courses can be linked to each other, facilitating implementation of a networked curricular model. The most appropriate elective courses and modules for each learner are determined by recommendation algorithms, as outlined earlier in this paper.

PERCEPOLIS is composed of three major components: i) a multi-database system that stores, integrates, and retrieves learning artifacts; ii) a context-aware recommender system responsible for identification of the most appropriate and beneficial learning artifacts for each learner, based on context that includes the learner's needs and interests; and iii) an intelligent multi-agent system. We articulate the details of ii) and iii) in the remainder of this section.

I. Recommendation Algorithms

The focus of content-based filtering techniques is solely on identifying resources that are similar to what learners have accessed in the past. This complicates the recommendation of new learning artifacts. Collaborative filtering techniques consider only similarities among learners' rating information, and as a result neglect content-based relationships among resources [8, 19]. To alleviate these shortcomings, we employ a combination of content- and collaborative-based filtering techniques in designing recommendation algorithms for PERCEPOLIS.

Two types of contextual information are utilized:

- **Explicit contextual information:** Provided directly by the learner or institution by completing surveys. This information can be classified into four categories:
  a. **Learner profile:** including academic records (list of courses and modules passed, grades, GPA, target degree, major, etc.), and personal profile (location, disabilities, interests, needs and skills).
  b. **Module profile:** including information such as prerequisites, contents (by topic and learning artifact), and author.
  c. **Instructor profile**: including a list of courses taught, skills, research interests, etc.
  d. **Environment profile:** including information about the institution and facilities; e.g., list of laboratories, disability accommodations, and computing facilities.
- **Implicit contextual information:** This information is inferred, and falls into one of two categories:
  a. **Learner tacit profile**: such as learning style; learner's infrastructure (device, operating system, networking); access records; tacit skills, e.g. passing a certain module may enable a new skill; skill level; e.g., amateur or professional; tacit interests, e.g.

passing a certain module with high grade may reflect the learner's interest in that topic.
  b. **Module tacit profile:** such as level of difficulty (inferred from the grades), audience (based on frequency of use in specific courses, or learners who have taken the module).

PERCEPOLIS includes algorithms for the following tasks:

- **Recommending the *N* most appropriate courses for the learner:** The algorithms for recommending courses offered in and outside of the learner's department, respectively, are depicted in Figure 2.
- **Recommending the *N* most appropriate elective modules for each course selected:** The algorithm for selecting elective modules is depicted in Figure 3. For mandatory modules, the system needs to check whether the learner has already passed the mandatory module. If the learner is interested in learning more about the topic, the system will recommend follow-up modules that can be taken for extra credit. These follow-up modules can be identified with the help of the algorithm proposed for elective modules, described later in this paper.

The interests and needs of a learner may change in the course of his or her perusal of learning artifacts. PERCEPOLIS recognizes this dynamism by providing updated recommendations in the course of the learning process, based on the progress of the learner in the target course. Moreover, the content of the selected modules is updated based on the learner's computing infrastructure, which includes bandwidth, access device, etc.

Another important issue considered in recommendation algorithms is local autonomy and heterogeneity of the databases housing the learning artifacts. The search routine depicted in Figure 4, and used by the recommendation algorithms in Figure 3, is based on the *Summary Schemas Model (SSM)*. SSM was developed to provide linguistic support for automatic determination of semantic similarity between different access terms [27]. The SSM leverages specific linguistic relationships between schema terms to create a hierarchical global data structure that describes the availability of information in all local databases. The semantic relationships used in the SSM taxonomy are synonymy and hypernymy/hyponymy. The hypernym of a word is a term with a more general meaning, such as *cat* for *lion*. The opposite relationship - a more specific meaning, is denoted as hyponymy, such as *tiger* for *cat*. Synonym links are symmetrical. Hypernym and hyponym links are reciprocal.

The SSM taxonomy used by the search algorithm is based on the concept of a *federated database system,* which is defined as a collection of cooperating database systems that are autonomous and possibly heterogeneous [28]. In the proposed hierarchy, leaf nodes are *module schema,* such as the "Random Access Memory" module; and higher nodes are *course schema,* such as the "High Performance

Computer Architecture; *domain schema,* such as "Computer Architecture"; *department schema,* such as "Computer Science"; and finally, *institution schema,* such as "Missouri University of Science and Technology".

II. Intelligent Software Agent Model

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

More specifically, when a learner, L, is required to take a course, C, an agent, D_{LC} is created to represent the learner in that course. D_{LC} gathers explicit context information about the learner's profile and infers implicit context information about the learner's academic background, interests, needs, and skills. D_{LC} recommends the top N elective courses to the learner with the help of the recommendation algorithm depicted in Figure 2. Moreover, after the learner selects her/his desired courses, then D_{LC} (with the help of the recommendation algorithm shown in Figure 3 and interaction with the agents of course modules) identifies a personalized trajectory of mandatory/elective modules and learning artifacts for the learner. Furthermore, D_{LC} ensures that the learner peruses required material - notes, sample programs, exercises, and the like. D_{LC} also alerts the learner to timelines, class schedules, learning/discussion schedules, project deadlines, appointments with the instructor, and corresponding preparations. The instructor agent for the course, D_{IC}, ensures that the learner meets all requirements for each mandatory module, and collaborates with D_{LC} to ensure that the learner is supplied with all required course material. D_{IC} also informs the instructor about the progress of the learner and alerts the instructor whenever the learner is progressing too slowly or too rapidly.

Conclusion

In this paper, we introduced PERCEPOLIS, a pervasive learning cyberinfrastructure that facilitates self-paced personalized learning. We proposed a context-based recommender system that utilizes a combination of content-based and collaborative filtering techniques to determine the most appropriate and beneficial educational artifacts for each learner, based on a wide array of learner attributes and environmental considerations. We also described three types of intelligent software agents that execute these algorithms: 1) learner agents, 2) instructor agents, and 3) course agents. The agents communicate with each other and use the services provided by recommender systems (such as our proposed recommendation algorithms) to personalize a learning trajectory for the learner and manage the learning process by checking his or her progress.

Extensions to this research planned for the immediate future include enhancement and predictive modeling of the recommendation algorithms for performance and accuracy, and implementation of a complete prototype of the cyberinfrastructure.
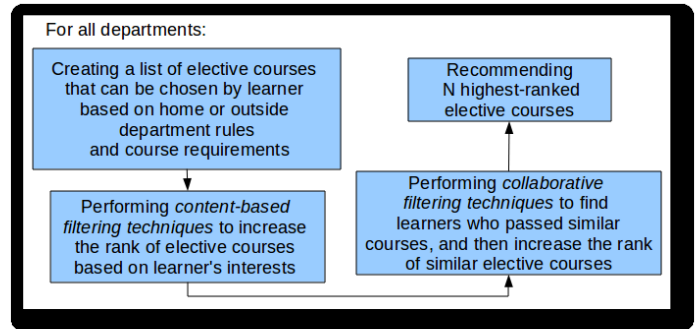

FIGURE 2
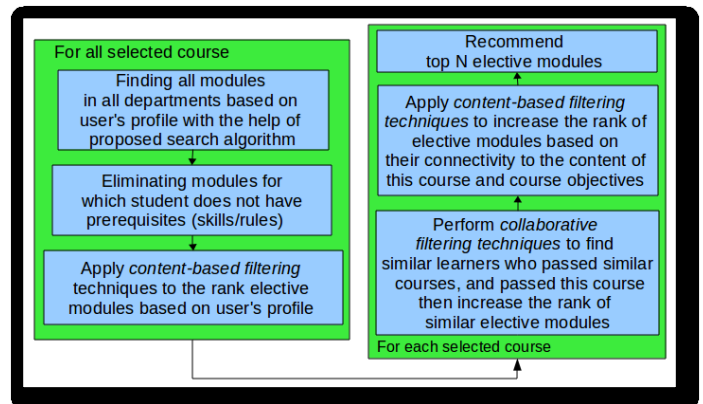RECOMMENDATION ALGORITHM FOR ELECTIVE COURSES

FIGURE 3
RECOMMENDATION ALGORITHM FOR FINDING THE N MOST APPROPRIATE ELECTIVE MODULES FOR SELECTED COURSES
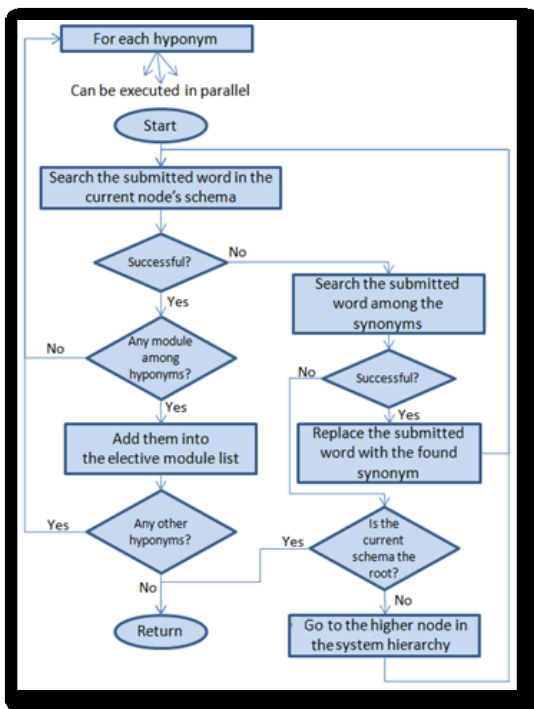
FIGURE 4
PROPOSED SEARCH ALGORITHM (BASED ON SUMMARY SCHEMA MODEL)

ACKNOWLEDGMENT

REFERENCES

[1] Jones, V. and Jo, J.H. "Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology". In R. Atkinson, C. McBeath, D. Jonas-Dwyer and R. Phillips (Eds), Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference, pp. 468-474. 2004.

[2] Grimm, R., Davis, J., Lemar, E. and Bershad, B. "Migration for pervasive-applications".
http://cs.nyu.edu/rgrimm/papers/migration02.pdf. 2002.

[3] Hurson, A. R. and Sedigh, S. "PERCEPOLIS: Pervasive cyberinfrastructure for personalized learning and instructional support". Intelligent Information Management, 2, pp. 583-593. 2010.

[4] Graf, S. and Kinshuk. "Adaptivity and Personalization in Ubiquitous Learning Systems". HCI and Usability for Education and Work. Springer, 5298, pp. 331-338. 2008.

[5] Liu, Z. and Liu, Y. "An adaptive personalized E-learning model based on agent technology". *WTOS* 7, 12, 1443-1452. December 2008.

[6] Jun, SJ., Han, SK., Kim, SH., Kim, HC. and Lee, WG. "A context model based on multi-agent in U-learning environment, Edutainment'07, Vol. 4469, pp. 274-282. 2007.

[7] Dey, A. K. and Abowd, G. D. "Towards a Better Understanding of Context and Context-Awareness". GVU Technical Report. GIT-GVU, pp. 99-22. 1999.

[8] Luo, J., Dong, F., Cao, J. and Song, A. "A context-aware personalized resource recommendation for pervasive learning". *Cluster Computing* 13, pp. 213-239. June 2010.

[9] Marivate, V. N., Ssali G. and Marwala, T. "An Intelligent Multi-Agent Recommender System for Human Capacity Building". CoRR08, pp. 909-915. 2008.

[10] Lange, D. and Oshima, M. "Programming and developing Java mobile agents with Aglets. Reading". MA: Addison-Wesley. 1998.

[11] Sakthiyavathi, K. and Palanivel, K. "A generic architecture for agent based E-learning system". IAMA'09, pp. 1-5. 2009.

[12] Garruzzo, S., Rosaci, D. and Sarne, G. M. L. "ISABEL: A Multi Agent e-Learning System That Supports Multiple Devices". IAT'07, pp. 485-488. 2007.

[13] Graf, S., MacCallum, K., Liu, T., Chang, M., Wen, D., Tan, Q., Dron, J., Lin, F., Chen, N., McGreal, R. and Kinshuk. "An Infrastructure for Developing Pervasive Learning Environments", Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 389-394. 2008.

[14] Dorça, F. A., Lopes, C. R. and Fernandes, M. A. "A multiagent architecture for distance education systems". In: III IEEE International Conference on Advanced Learning Technologies, Atenas, Grecia, pp. 368-369. 2003.

[15] Alexakos, C. E., Giotopoulos, K. C., Thermogianni, E. J., Beligiannis, G. N. and Likothanassis, S. D. "Integrating E-learning Environments with Computational Intelligence Assessment Agents". Proceedings of World Academy of Science, Engineering and Technology, V. 13. May 2006.

[16] Capuano, N., Marsella, M. and Salerno, S. "ABITS: An Agent Based Intelligent Tutoring System for Distance Learning". Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems, Montreal, Canada. 2000.

[17] Jiao, Y. and Hurson, A. "Performance analysis of mobile agents in heterogeneous mobile database systems," in Proceedings of the IEEE International Conference on Emerging Technologies (ICET'03). 2003.

[18] Schilit, B. N., Hilbert, D. M. and Trevor, J. "Context-aware communication," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 46–54. 2002.

[19] Su, X. and Khoshgoftaar, T. "A survey of collaborative filtering techniques". In Advances in Artificial Intelligence Volume 2009. Number 2009.

[20] Schilit, B., Adams, N. and Want, R. "Context-aware computing applications". In: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pp. 85–90.

[21] Adomavicius, G. and Tuzhilin, A. "Context-aware recommender systems". In F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors, Recommender Systems Handbook, Springer Verlag, pp. 217-253. 2010.

[22] Liu, T. and Chu, Y. "Using ubiquitous games in an English listening and speaking course". Impact on learning outcomes and motivation. Computers & Education, 55(2), pp. 630-643. 2010.

[23] Ogata, H. and Yano, Y. "Context-aware support for computer-supported ubiquitous learning". In: Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education. 2004.

[24] Vladoiu, M. and Constantinescu, Z. "U-Learning within a Context-Aware Multiagent Environment". International Journal of Computer Networks & Communications (IJCNC), Vol. 3, No. 1, pp. 1-15. January 2011.

[25] Rogers, Y., Price, S., Randell, C., Fraser, D. S., Weal, M. and Fitzpatrick, G. "Ubi-learning integrating indoor and outdoor learning experiences". Communications of the ACM, 48(1), pp. 55–59. 2005

[26] El-Bishouty, M. M., Ogata, H. and Yano, Y. "PERKAM: Personalized knowledge awareness map for computer supported ubiquitous learning". Educational Technology and Society, 10(3), pp. 122–134. 2007.

[27] Bright, M.W., Hurson, A.R. and Pakzad, S. "Automated resolution of semantic heterogeneity in multidatabases". ACM Transactions on Database Systems, pp. 212-253. 1994.

[28] Sheth, A. and Larson, J. "Federated database systems for managing distributed, heterogeneous, and autonomous databases". Computing Surveys, 22(3). 1990.