Real-Time Teacher Assistance in Technologically-Augmented Smart Classrooms

Georgios Mathioudakis*, Asterios Leonidis*, Maria Korozi*,
George Margetis*, Stavroula Ntoa*, Margherita Antona*, and Constantine Stephanidis*†
*Institute of Computer Science, Foundation of Research and Technology – Hellas (FORTH),
Heraklion, GR-70013, Greece
Email: {gmathiou}{leonidis}{korozi}{margetis}{stant}{antona}{cs}@ics.forth.gr
† Department of Computer Science, University of Crete

Abstract*—The role of the teacher in the classroom environment is of crucial importance for the effectiveness of the learning process. However, recent studies on the technological enhancement of education have shown that teacher's activities are not adequately supported, as the focus remains rather on the student's side. This article discusses a learner-centric approach towards supporting instructors in improving the teaching and learning processes in ambient educational environments. The proposed solution introduces an intelligent multi-agent infrastructure that monitors unobtrusively the students' activities and identifies potential learning weaknesses and pitfalls that need to be addressed at an individual or classroom level. Such real-time insights enable the instructor to intervene providing help and adapt the teaching process according to the needs of the class. For that to be achieved, several applications have been developed: (i) a real-time classroom activity visualizer, (ii) a behavioral reasoner that aims to identify common behaviors by analyzing classroom activities, (iii) a statistics records manager targeting to showcase students' progress and performance at both an individual and classroom level, and finally (iv) a series of mini-tools that enhance typical procedures that can be found in conventional classrooms, such as the classroom attendance record, the schedule manager, etc. Following the system's description, findings of a preliminary expert-based evaluation are presented and some concluding remarks regarding the deployment of the system in real-life environments are formulated. Finally, potential future extensions of the system are proposed.

Keywords–ambient intelligence, education, smart classroom, teacher assistance, student monitoring.

I. INTRODUCTION

This article provides an extended version of the work [1] reported in The Fifth International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2013) in Nice, France. In this article, the specification and implementation of an embedded system targeted to support instructors during the educational process are further elaborated and discussed.

Ambient intelligence (AmI) is an emerging technological paradigm that defines sensitive digital environments that monitor their surroundings through pervasive sensorial networks and automatically adapt to facilitate daily activities [2], [3]. According to the Ambient Intelligence vision, digital systems provide user-interfaces embedded in the actual living space, enabling intuitive and natural interaction. AmI initially benefited mainstream areas such as home and office automation [4]. During the past few years though, remarkable efforts have been made towards applying AmI in a variety of domains such as education[1], health[2], entertainment[3] and many more.

The potential of AmI in education led to the introduction of the notion of "Smart Classroom". According to this, typical classroom activities are enhanced with the use of pervasive and mobile computing, artificial intelligence, multimedia content and agent-based software [5]. As a result, traditional artifacts such as desks and whiteboards are replaced by technologically enhanced equivalents aiming to support the learning process. The current realizations of the Smart Classroom vary, covering a wide range of topics. The most prevalent of them include applications for automatic adaptation of the classroom environment according to the context of use [6], [7], automatic capturing of lectures and teacher's notes [8], [9], enhancement of the learner's access to information and personalization of the classroom's material [10] and finally, supporting collaboration among classroom participants [11]. However, the majority of current research approaches focus on the learner's activities, without paying much attention to the role of the teacher.

During the learning sessions in a classroom the teacher duties, among others, include: (i) implementation of a designated curriculum, (ii) maintenance of lesson plans, (iii) assignment of tasks and homework, (iv) performance monitoring, and most importantly, (v) assistance provision when necessary. In general, curriculum activities outweigh monitoring and assistance tasks, especially in crowded classrooms (e.g., more than 20 students). Therefore, to enable effective and personalized tutoring, an automated method that observes students' behavior and identifies common problems is needed [12].

Towards this end, a tool named AmI-RIA (Real-time Instructor Assistant) is introduced in this article, aiming to support the teacher in the context of a learner-centric, ambient intelligence classroom. AmI-RIA monitors and analyzes students' activities in real-time so as to identify potential difficulties, either at an individual or at a classroom level, and notify the teacher accordingly (through the teacher's front-end application). The teacher can therefore concentrate on the lecture and rely on the system to monitor the classroom and prompt for intervention only when necessary (e.g., a student is out of task or performed poorly in a quiz). In addition to real-time monitoring, AmI-RIA offers a performance analysis

[1] AmI Playfield: http://bit.ly/1eM8EWL (Online: 5/2014)
[2] Ambient Intelligence for e-Health: http://bit.ly/1d7jDh6 (Online: 5/2014)
[3] Be There Now!: http://bit.ly/NGPxYb (Online: 5/2014)

tool that provides extensive metrics of students' progress and performance (based on previously collected data) that the teacher can use to either identify topics that require further elaboration or adapt the teaching methodology. Finally, AmI-RIA integrates tools that automate common classroom procedures, like attendance record keeping, quiz assessment and preparation of lesson's curriculum.

The rest of the paper is structured as follows. Section II presents related work on student monitoring in real classrooms and e-learning environments. Section III provides a description of the AmI-RIA system design along with the surrounding Smart Classroom environment. Sections IV and V present the system's implementation details. Section VI reports the evaluation results. Section VII discusses the challenges of a real-world deployment and finally, Section VIII and IX summarize the described work and highlight potential future enhancements.

II. Related Work

The widespread use of ICT (Information and Communication Technology) in learning environments has urged researchers to take advantage of the presence of technological equipment inside classrooms in order to enhance the learning and teaching process. Towards this objective, various intelligent systems that monitor students' activities and report valuable insights to the teacher have been developed, aiming to enhance both real and virtual (i.e., e-learning environments) classrooms.

A. Student Monitoring in Real Classrooms

In [13], the authors introduce Retina, a tool targeted to assist instructors that offer computer science courses to improve their curriculum by reporting the difficulties that students are facing during programming. Retina collects information about students' programming activities, such as attempts to compile their project, compilation and run-time errors, time spent for each assignment, etc. Retina logs past information about students' activities and generates informative reports both for them (i.e., self-evaluation) and the instructors. A live monitoring mechanism enables instructors to get insights for the programming session at run-time, so as to either address issues immediately during a lecture or adjust forthcoming assignments. An additional tool is also included that provides suggestions to the students, based on the collected data, via instant messages.

In [14] it is argued that teachers working in robotic classes have problems in keeping track of students' activities. As the authors claim, the real challenge for the instructors is to know when and how to intervene. Thus, they propose a system that collects data from the robotic environment and inform the teacher about the activities with which students are engaging and how they are progressing. The design of the system relies on the LeJOS programming platform for Lego Mindstorms, where two agent modules are used for data collection, one embodied into the robot and the other deployed in the programming environment.

Another monitoring system targeting programming courses is presented in [15]. The authors envision a system capable of detecting students' frustration, at a coarse-grained level, using measures distilled from student behavior within a learning environment for introductory programming. The monitored data include compilation errors, error messages, source code and other relevant information. As they argue, frustration is potentially a mediator for student disengagement. Thus, detecting it will assist instructors to intervene in ways that will help students remain motivated. Following the same pattern, past [16], [17] and recent studies [18], [19], focus on monitoring student behavior during programming courses in order to generate insights that will assist instructors improve the learning experience.

MiGen [20] is a related intelligent environment designed to support 11-14 year-old students in learning algebraic generalizations. The system aims to assist the teaching process by informing teachers of students' progress, the appearance of potential misconceptions and disengaged students. As the authors claim, this will allow teachers to provide learning in a personalized way. To fulfill this task, MiGen visualizes the students' progress based on their attainment of specific landmarks as they are working on mathematics generalization tasks. MiGen was one of the first to introduce a classroom overview panel to serve the teacher's needs, however remained at a very basic level in terms of the amount of information displayed and the user-interface quality.

A relevant, extended study is presented in [11], [21], [22]. The proposed system, named I-MINDS, consists of a group of intelligent agents that are able to track the activities and progress of students. This tracking mechanism targets to identify any problematic situations that may occur to students and then inform the instructor or assist the student to overcome the problem. The I-MINDS system offers rich insights of students' behavior to the teacher, however, it is mainly focused on the social aspect of the educational process and intended mostly to assist the formation of collaborative groups inside the classroom environment.

The aforementioned systems can partially provide real-time information to the instructor, however, they share some major drawbacks: (i) they are usually targeted to specific contexts of use (e.g., programming courses), (ii) they offer rather poor user interfaces, in terms of usability, that hinder information extraction, and, (iii) they bind the students on using actual computer machines during the educational process.

B. Student Monitoring in E-learning Environments

The Smart Classroom notion usually refers to real classrooms. However, a fair number of studies exist that aim to support instructors within e-learning environments through student monitoring. Some of them also introduce innovative methods of visualizing the students' activities during the educational process.

In [23] and [24] a web-based environment is proposed, capable of collecting students' traces produced by their interactions in order to visualize the virtual classroom. Due to the web-based nature of the system, the provided visualization helps the teacher control and interact with the classroom. Participants are represented by Chernoff faces [25], whose facial characteristics evolve over time according to their activities. Additionally, the system represents the pedagogical activities as bubbles, which grow or shrink proportionally to the number of the participants.

In [26] and [27] a relevant system is presented, named CourseVis, which is capable of generating graphical representations of what is happening in the classroom by analyzing students' activities data collected in the context of a CMS (Course Management System). CourseVis creates a number of plots in order to visualize social, cognitive and behavioral aspects of the learners. Furthermore, it includes a mechanism for viewing statistical data from students' interactions, such as the number of accesses to each resource, the history of pages visited, etc.

Likewise, [28] presents an intelligent agent system that supports teachers in supervising learners in LAMS (Learning Activity Management System). The system is capable of notifying the instructor for common problems about participation and contribution of students during educational activities. However, for that to be achieved, the instructor is required to determine expectations for the attendance and contribution of the learners for each activity. These expectation parameters include the typical execution time, the contribution level on collaborative activities, the expected score, etc. Finally, a notification agent is used to deliver messages and information to the supervisor of the lesson and to the learners as well.

The systems discussed above constitute representative state-of-the-art approaches in the domain of student monitoring in e-learning environments that aim to assist instructors. However, several drawbacks can be identified in these solutions. On the one hand, the user-interfaces at the teacher's endpoint although they are more expressive and informative than the ones of the real-classroom systems, cannot be considered intuitive and the information extraction still remains a tough task. For instance, Chernoff faces are a useful tool for indicating student inactivity, but in more complex situations (e.g., progress and performance tracking) their expressiveness is limited. On the other hand, the e-learning environments studied do not offer an effective real-time assessment method, which is required in intelligent learning environments [29].

Thus, there is a clear need for a system that can: (i) be deployed and operate in real classrooms, (ii) monitor unobtrusively the students through their interactions taking place during the educational process, (iii) produce valuable insights about their behavior in real-time, and finally, (iv) deliver those insights to the teacher through an intuitive, yet rich, user interface.

III. System Design

The AmI-RIA system proposed in this paper aims to bridge the gap between students and teachers by providing valuable insights to the latter. For that to be achieved, several smart systems are precisely coordinated and tightly collaborate to shape the Smart Classroom (depicted in Figure 1), as it is envisioned and implemented in the context of the FORTH-ICS (Foundation for Research and Techonology Hellas - Institute of Computer Science) AmI Programme[4] (an interdisciplinary RTD Programme aiming to develop and apply pioneering human-centric AmI technologies and Smart Environments).

Figure 1: The Smart Classroom simulation space at FORTH-ICS AmI Facility. Teacher Assistant is installed in the teacher's PC visible on the left.

As depicted in Figure 2, the ClassMATE system [30] forms the backbone infrastructure that supports the intelligent classroom. ClassMATE monitors the Ambient environment and is capable of making context-aware decisions in order to assist the student in conducting learning activities. Furthermore, it is responsible for orchestrating the various artifacts that can be found in the classroom, for example, the augmented desk (Figure 3(a)) and the SMART Board[5] (i.e., the commercial interactive whiteboard depicted in Figure 3(b)). In more details, the augmented desk [31] is an enhanced school desk that uses computer vision technology to recognize books and book pages in order to provide physical and unobtrusive interaction without requiring any special device [32]. The SMART Board supports the educational tasks by offering a shared interactive area that extends the augmented student desks as applications can seamlessly migrate among them. For instance, if the teacher asks a student to start answering the questions of an exercise in front of the class, the achieved progress will be automatically transferred back to the desk when done.

A. Overall Architecture of AmI-RIA

The primary goal of AmI-RIA is to inform the teacher about the students' activities and identify potential weaknesses by monitoring their interactions and generate classroom-wide progress and performance metrics. Towards this objective, a distributed architecture (Figure 4) is introduced that consists of two major components. The first component is an intelligent

[4]http://www.ics.forth.gr/ami (Online: 5/2014)

[5]SMART Board: http://bit.ly/1n5RGKz (Online: 5/2014)

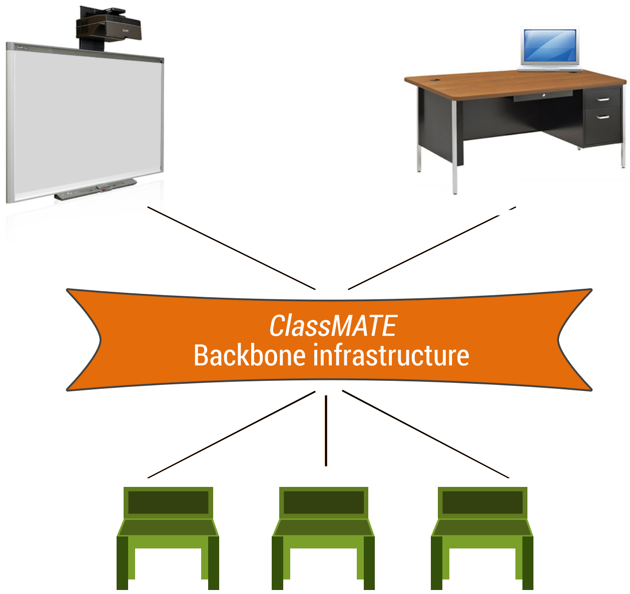
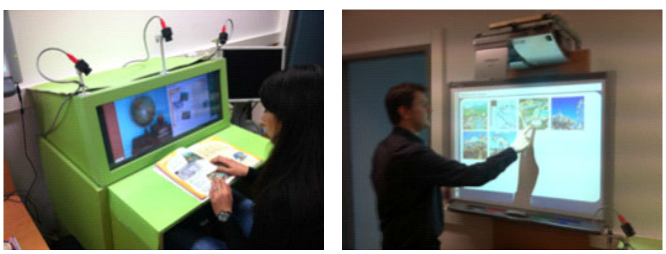


Figure 2: The Smart Classroom prototype as implemented at FORTH-ICS



(a) Augmented Desk Prototype v1.0 (b) SMART board

Figure 3: Artifacts of the Smart Classroom environment

agent deployed on the students' desks, named Desk Monitor, which monitors each individual student's interaction. The second major component, the Teacher Assistant, is an intuitive frontend application deployed at the teacher's desk to facilitate the visual representation of the monitoring data (i.e., classroom overview) and simplify classroom control, such as assignment submission, exam distribution, etc.

Desk Monitor agents collect the monitoring traces that students generate when working on their desks and through a reasoning process draw conclusions about their behavior. Both the collected and the inferred knowledge is transmitted in real-time to the Teacher Assistant application, which is responsible to present them appropriately (e.g., highlight inactive students, prompt teacher action, etc.). Data exchange is performed through a generic services interoperability platform, named FAmINE (FORTH's AMI Network Environment), presented in [33]. FAmINE provides the necessary functionality for the intercommunication and interoperability of heterogeneous services hosted in AmI environments. It encapsulates mechanisms for service discovery, event driven communication and remote procedure calls.

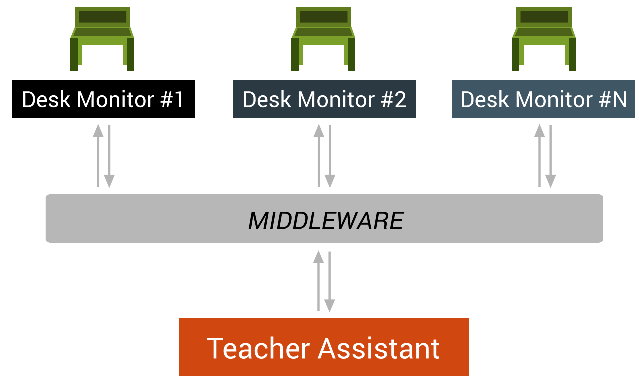


Figure 4: Distributed architecture of AmI-RIA

B. Data Collection

The collection of the monitoring data originating from the students is achieved through the classroom's backbone infrastructure and the aforementioned "smart" artifacts (i.e., smart desks and boards) that appear in the classroom environment. The augmented desk, the most important artifact of the classroom and the main source of monitoring data, is equipped with an interactive learning environment named PUPIL [34]. In short, the PUPIL framework facilitates the design, development and deployment of pervasive educational applications. Using this framework, several applications were developed and deployed in the Smart Classroom environment as already presented:

- ClassBook application: Digitally augments physical books by introducing interactive alternatives to printed elements like images, exercises, etc.
- Multimedia application: Multimedia content exploration and display (e.g., images, videos)
- Dictionary application: Displays textual and/or multimedia information about a topic or word
- Multiple-Choice exercise application: Digital alternative to the classic multiple-choice quiz. Questions are further enhanced with help buttons offering hints to learners
- Hint application: Gradually assists students towards finding the right answer by offering personalized hints. Supports the development of critical thinking skills

The aforementioned applications provide the required data source by exposing any detected interaction of students to the backbone infrastructure of the Smart Classroom. Some of the activities of interest for the AmI-RIA system include:

- login when a student sits on a desk
- course book page fanning
- launch of an exercise session
- answer submission
- use of contextual help provided by the learning system
- browsing and sharing of multimedia galleries

These activities along with the related data become available to the Desk Monitor agents by ClassMATE through a FAmINE-enabled bridge interface. For creating such interfaces FAmINE supports IDL (Interface Description Language), a specification language used to describe a software component's interface. IDL addresses interoperability issues as it

```
{
?student      a                               student:Student.
?student      student:Current_Exercise        ?exercise.
?exercise     student:HasQuestion             ?question.
?question     student:Hints_Used_For_Question ?hints.
?hints        math:greaterThan                "2"^^xsd:int.
}
=>
{
?question     student:Hints_Overuse           "true"^^xsd:boolean.
}.
```

Figure 5: Example of a N3 rule

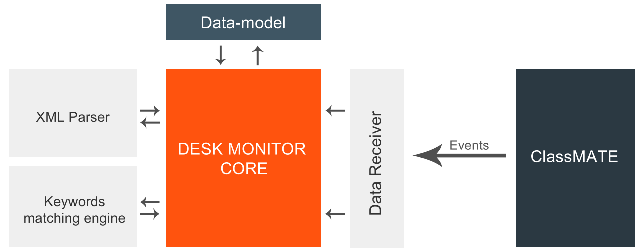


Figure 6: Anatomy of a Desk Monitor agent

is language-independent enabling this way communication between components developed in different languages (e.g., Java, C#). The defined interface includes structures, events and remote procedure calls, used for the regular messages exchange between the classroom's backbone infrastructure and the Desk Monitor agents.

C. Data Management and Reasoning

Ontologies are widely accepted as a tool for modeling contextual information about pervasive applications [35], as they address the problem of data heterogeneity between applications and support data interconnection using external popular vocabularies, such as FOAF [36] and Dublin Core metadata [37]. Furthermore, they also enable knowledge inference using semantic reasoners whose rules are implemented by means of ontologies.

AmI-RIA makes extensive use of ontologies. An RDFS schema [38] has been implemented that defines classes for the relevant entities (e.g., Teacher, Student, Book) and the activities (e.g., Open book, Start exercise) that can potentially take place in a classroom environment. Additionally, a set of taxonomies has been defined, based on RDFS properties, to associate classes and create activity hierarchies (e.g., Submit_Exercise isA Student_Act). Collected data are stored internally in the form of RDF triplet statements following the defined schema.

The required RDFS hierarchies for the entities were implemented using Protege, an open source ontology editor and knowledge base framework described in [39]. Protege offers a suitable environment for modeling the entities and shaping SPARQL [40] queries.

The reasoning process of the AmI-RIA system is supported by the SemWeb library for .NET [41]. SemWeb supports SPARQL queries for information retrieval over the data and incorporates the Euler engine [42], a popular backward chaining inference engine. The rules used by the Euler engine are written in external files using the Notation3 syntax [43] (Figure 5), an RDF syntax designed to be human friendly. Rule decoupling facilitates system maintenance and scalability, as the insertion of new rules or the modification of existing ones can be done without affecting the core of the AmI-RIA system.

IV. Desk Monitor Agents

The Desk Monitor agents constitute the core components of the AmI-RIA system, as they execute the inference rules over the collected interaction data to identify potential troublesome situations (e.g., inactive or off-task behavior, problems in understanding of concepts, etc). To that end, the agents apply a goal-driven reasoning process on contextual knowledge through a backward chaining inference engine (i.e., Euler), to identify such alarming situations, semantically defined using taxonomies, inside the classroom environment.

A. Architecture

As depicted in Figure 6, a desk monitor agent consists of five major components, namely, Data Receiver, Data-model, XML Parser, Keywords-matching engine and the Core.

The backbone infrastructure of the Smart Classroom packs the data collected by the student desks and the SMART Board artifact into events and transmits them through the middleware to a Desk Monitor module. The *Data Receiver* component provides the handlers for listening to such events and is responsible for forwarding the data included in the events to the core of the agent. The latter follows a procedure of forming a *Data Model* of the student based on the monitored interactions, which are then stored internally using RDF triplets. Since the students may switch desks occasionally, no permanent storage exists for the RDF data-model constructed by the Desk Monitor; data are erased at the end of each session. A set of predefined rules is evaluated upon the data-model to infer new knowledge.

An exercise or test in the electronic form of the book presented on the augmented desk is implemented following a custom XML schema. Each exercise file contains, among other data, the type of activity (e.g., multiple-choice), the questions, the available answers for each question, etc. For this to be parsed, Desk Monitor uses a *XML Parser* component that is capable of reading the exercise files and instantiate internal data structures.

For each learning session, a set of relevant material, books and pages is defined and checked against the material that a student has opened on the desk. Thus, it is possible to identify whether a student has opened some non-relevant material (i.e., off-task) just by comparing book titles and page numbers. Although this method is safe and effective, it has the drawback of requiring all the educational material to be analyzed and categorized accordingly at any time. To address this issue, the *Keywords Matching* component is implemented to enhance the aforementioned process by matching semantically the content appearing on a page to that of the learning session. To do so, the component exploits the metadata and text appearing on the materials used in the classroom. The procedure of the keyword matching consists of two phases, one for indexing the keywords and text from the listed relevant pages and one for matching the current's page keywords against the indexed ones.

B. Behavioral monitoring

The Desk Monitor uses the defined taxonomies and the semantic rules and produces insights about the behavior of a student in the classroom, as soon as student activity is detected. The list of situations that can be currently detected includes: (i) off-task students, (ii) inactive students, (iii) students that face difficulties during exercise solving, (iv) students that face difficulties during exercise submission and, (v) students that misuse the contextual-help of the learning system. A detailed description for each situation, along with its importance within the educational context, is provided in the following sections:

1) Off-task: According to Caroll's Time-On-Task hypothesis [44], the longer students engage with the learning material, the more opportunities they have to learn. Therefore, if students spend a greater fraction of their time engaged in behaviors where learning is not the primary goal, they will spend less time on-task and as a result learn less. In [45], the authors argue that off-task behavior indeed has a negative impact on students' performance and investigate different types of off-task behavior. The evaluation results of the aforementioned study indicate that the frequency of off-task behavior is a good predictor of the students' performance, though, different types of off-task behavior result in different negative correlation to learning. Furthermore, in [46] authors also state that off task behavior appears to be associated with poorer learning performance at an aggregate level. To identify off-task students, the AmI-RIA system checks the material displayed on a student's desk to determine if it is relevant to the topic discussed in the classroom based on the activity in hand. Thus, it examines whether (i) the currently opened book, (ii) the opened page and (iii) the content of the currently opened page semantically belong to the current activity.

2) Inactivity: During classroom activities, especially exercise solving, it is common for students to start working on a task and after a while give up because they get bored or distracted. Inactivity is defined as a type of off-task behavior where the student does not interact with the learning object at hand at the appropriate time. According to [45] and [47], inactivity indicates that a student is disengaged with a certain task and can be used as a quite accurate performance predictor. AmI-RIA exploits the typical learning time describing the amount of time that a student is expected to work with or through a learning object [48], to specify if and when a student's interaction is taking too long to be executed. For that to be achieved, AmI-RIA gets notified by ClassMATE about the actions that a student performs when interacting with a learning object (e.g., an exercise, a text passage, etc.). However, since not all the students interact with the exercise at the same pace, individual factors should be taken into account. Towards this, the learning level of a student is estimated, based on the average score in related activities, and then used as a bias parameter in the formula calculating the total interaction time.

3) Weaknesses during problem solving: The PUPIL framework offers personalized tutoring in the form of contextual help (i.e., hints) for each question of an exercise in order to help students find the right answer. Hints gradually increase the amount of help provided, thus a student using the last hint takes advantage of all the available help. AmI-RIA monitors the amount of help asked and the selection made afterwards to calculate the student's performance. In case a student uses the maximum amount all available hints, but still does not answer correctly, then the system marks that the student has difficulties regarding this question and the concept it refers to.

4) Problems on exercise completion: Identifying whether a student faces difficulties during exercise solving is quite challenging, since a single pass/fail indicator does not always reveal the actual progress of a learner on a specific topic. To this end, instead of generalizing conclusions based merely on the score of the exercise in hand, the student's performance record on relevant topics/similar exercises is taken into consideration. Thus, even if the score is not a failing one, a potential weakness can be identified if there is a large decline on the score based on the student's record.

5) Misusing the Learning System: Sometimes students interact with exercises according to a set of non-learning-oriented strategies described in [45] and further studied in [49], known as gaming the system. Such strategies involve behaviors aimed at systematically misuse the help provided by the system in order to advance in exercise instead of actually making use of the material of the intermediate hints. A set of rules has been created to track students who repeatedly ask for help within a small time frame until they get the maximum one.

C. Extensibility of the Reasoning Mechanism

Deploying the AmI-RIA system in a real classroom may raise the need of adjusting the currently defined semantic rules or the creation of new ones to infer new knowledge. To address this issue, the semantic inference rules are defined in external files, completely separated from the system's logic. Furthermore, to enable even non-programmers (e.g., teachers) to edit or create new rules, the Notation3 RDF format is used, which is designed to be human friendly.

V. TEACHER ASSISTANT

AmI-RIA offers an intuitive front-end application deployed at the teacher's computer (or portable tablet device) named Teacher Assistant, through which the instructor can monitor at real-time via live feed the activities that take place in the classroom and identify potential weaknesses. For that to be achieved, every Desk Monitor Agent propagates the collected data and produced inferences through the classroom's middleware to the Teacher Assistant application, which is responsible for presenting them accordingly. By the time a student logs to a desk via an RFID (Radio-frequency identification) sticker appearing on books, the system pairs his/her unique id with that particular desk. Thus, every active desk in the classroom is bound to a specific student and can be uniquely identified.

In terms of data management, the Teacher Assistant makes use of RDF triplets and RDFS taxonomies, which extend the ones defined for the Desk Monitor agents. The implemented Classroom ontology defines entities appearing in the classroom such as courses, books, students, teachers, etc., as well as activities that represent the actions taking place. Indicatively, Figure 7 presents the defined hierarchy for activities involving

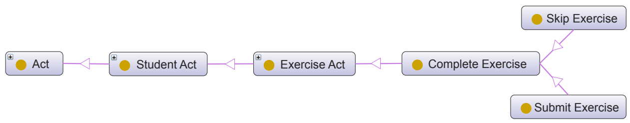


Figure 7: RDFS taxonomy for exercise activities

Figure 8: Classroom overview panel of Teacher Assistant

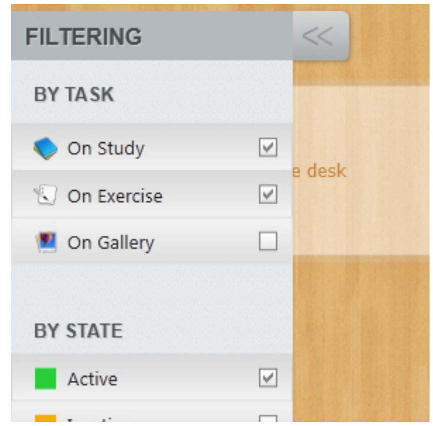


Figure 9: The filtering mechanism of the Teacher Assistant

exercise solving. As the semantic web notion prompts for more structured and connected data, two external popular vocabularies were used in the Classroom ontology, namely, the FOAF specification and the Dublin Core metadata. Using semantic web vocabularies allows intelligent agents make sense of the entities appearing in ontologies and the connections between them.

A. *Classroom Overview*

In terms of GUI, the Teacher Assistant (Figure 8) adheres to the natural mapping rule [50] that leads to immediate understanding because it takes advantage of physical analogies. As such, each student in the classroom is represented by a Student Card. Non-occupied desks are presented as semi-transparent empty cards, whereas the layout resembles the one of the physical desks. As a result, the teacher can easily locate a student in the classroom through the virtual class map or access the attendance record to see the absent students. The Teacher Assistant, following a responsive philosophy, adapts its layout accordingly to support devices with smaller screen resolution like tablets and portable computers.

B. *The Student Card*

The Student Card contains both personal information, such as the name and the profile picture, and also information regarding the current activities and status of the student. During the course, the student might be engaged with various activities such as reading a passage from a book, solving an exercise, browsing a multimedia gallery, etc. Providing specific details on such classroom tasks allows the teacher to be constantly informed about the students' attention levels and potential learning difficulties. To this end, each Student Card adjusts to represent the current learner's status at any given moment. For instance, when a student is reading, the card displays the book title and the respective page numbers; during an exercise, additional information is displayed regarding the topic, difficulty and the student's progress, finally when a student launches a multimedia gallery, a small set of relevant keywords is displayed on the card.

However, during a lecture the students might lose interest and deviate from the teacher's suggestions. This kind of information could ideally prompt the teacher to investigate the reasons of such attention lapses and try to maintain the student's interest. For that purpose, the Student Cards are enriched with visual cues (e.g., different border colors, self-explanatory icons) to mark on-task, off-task and inactive behaviors. Finally, since the implemented system targets large and crowded classrooms, the visual information may become too large to be handled easily. To overcome this difficulty, a filtering mechanism (Figure 9) that allows the teacher to focus on specific student groups is incorporated. For instance, during exercise time, the teacher can choose to only view inactive students.

C. *Assessment*

Exercises are considered to be a key aspect of the learning process in a classroom as through performance monitoring potential learning gaps can be revealed and the domains where the teacher should focus are highlighted. AmI-RIA ensures that the teacher will be able to watch students' progress during exercise sessions by adjusting the Student Card appropriately to display the exercise's name, the related topic and the student's current score. More detailed information about student's performance is available through two special-purposed windows.

The first one presents a detailed view of the aspects of the exercise at hand; in particular:

- type (e.g., multiple choice quiz, fill-in the gap, etc.)
- difficulty level (e.g., easy, medium, hard)
- typical learning time as defined in the LOM metadata

The second window (Figure 10) presents a complete log of student's actions regarding that exercise:

- number of answers given
- number of hints used per question

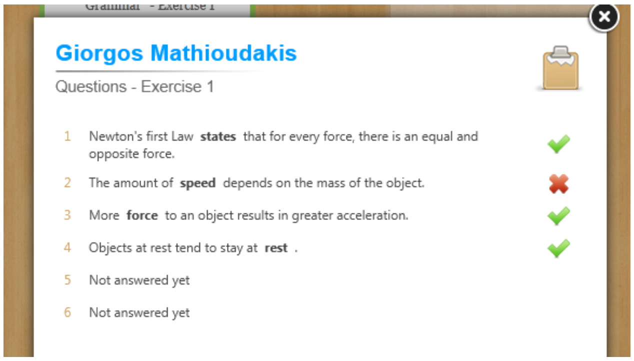


Figure 10: Exercise related interactions window

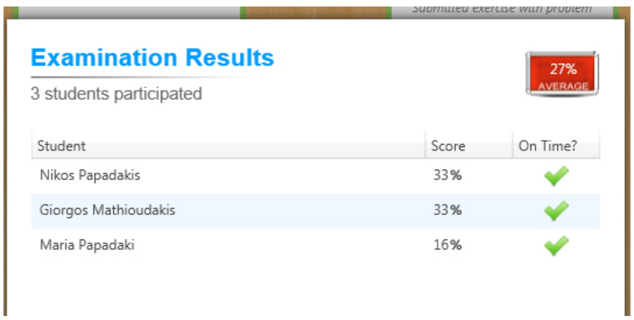


Figure 11: Test results window

- current score
- ratio of correct/wrong answers
- optionally, a problem indicator (if such decision was made by the respective Desk Monitor)

In addition to exercises, tests are also integral part of the educational process. Tests are a type of exercise where every student is obliged to answer and no help is provided. As soon as a test is initiated from the Teacher Assistant application, it automatically launches on every desk and deactivates the various assistive facilities (e.g., Thesaurus, Multimedia, etc.). During tests, the teacher can monitor students' progress as with common exercises and is able to request its immediate submission at any time. At that point, any tests that have not been submitted yet are automatically collected and a summarizing report (Figure 11) is presented with an average score for the entire classroom and a precise score for each student.

D. The Short Term Monitor

A teacher in the envisioned smart classroom is notified in real-time about the activities carried out by the students. However, eventually some notifications for a student will not catch the teacher's attention. Additionally, it is not practical to recall all the past notifications generated for one student. To address this issue, a mechanism is implemented for discovering trends in the student's activities and notify the teacher accordingly. For example, the action of a student to skip one exercise is not considered to be an important issue, however, if this student skips the fifth exercise in a row for the past few hours this means that there is a potential issue that the teacher should look after and intervene providing assistance. Each instructor can adjust the mechanism according to his needs (e.g., to be notified just for continuously inactive students); personal settings are stored on the teacher's profile in the system.

E. The Classroom Monitor

Individual statistics are automatically generated for each student by the respective Desk Monitor agent; however, accumulated metrics for the entire classroom are invaluable tools for teachers as through them behavioral patterns can be identified; an activity is considered to be a pattern if it is observed in a certain number of students in the classroom. For instance, if 85% of the students faced difficulties and performed poorly in an exercise, that may indicate that the exercise is too difficult and the teacher has to adapt the class' schedule to further elaborate on the related concepts. Similarly, if more that 80% of the students are off-task at the same time, then either a break might be helpful or the teacher should attract their attention and enhance their motivation. In any case, when AmI-RIA identifies a pattern, a special alert is generated to notify the teacher. The notifications appear on the top-right corner of the interface and can be dismissed with ease by the instructor.

F. Statistics

The data collected about the students' activities is used to build a rich history record, which is a vast source of semantic information based on the defined RDFS schema. This knowledge is exploited to generate statistics for the progress and performance of the students during short or long periods of time. Based on these statistics the teacher can identify the topics that need to be revisited or adapted and the thematic areas that seemed to have troubled each student in the past days, weeks, months, etc. Additionally, the generated statistics can be printed and handed-out to parents as an unofficial progress report for students.

The statistics component offers two alternative views, one at the classroom level and another for individuals (Figure 12). Both provide information about the overall performance, highlight topics in which the students achieved the highest and the lowest scores, and finally accumulate performance statistics per student and per lesson. More specifically, the individual statistics include: (i) average score per day, (ii) average out-of-context time per day, (iii) higher & lower score topics, and (iv) a lesson ranking based on student's score. The classroom-wide statistics include: (i) average score per day, (ii) out of context time per day, (iii) higher & lower score activities categorized by topic, and (iv) students' score ranking.

Usually in schools lessons for a classroom are taught by several teachers of different teaching professions (Mathematics, Physics, etc.). Therefore, a potential issue arise that the statistics view will also provide information that is not relevant or interesting to some specific instructor. For example, the Math teacher may not be interested in viewing statistics for the History lesson. A filtering mechanism has been incorporated in order to provide personalized views to the teacher values. Therefore, each teacher can choose to display statistics regarding only the lessons he is teaching or interested about.

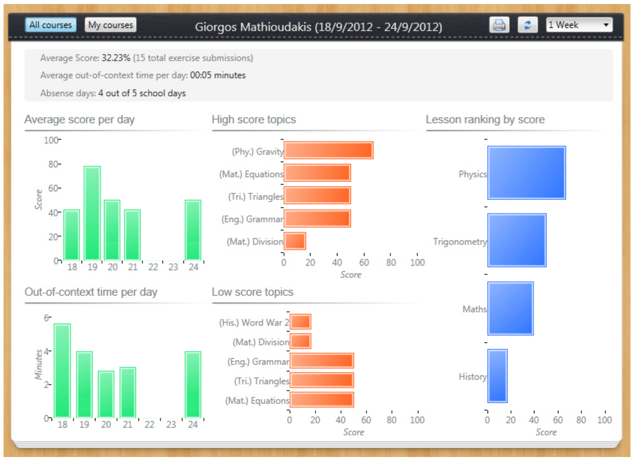


Figure 12: The individual statistics component

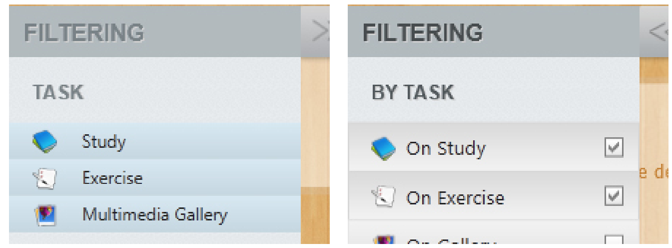
G. User authentication

The Teacher Assistant application displays sensitive information about the students' activities. Thereby, only authorized users must have access to the system. Additionally, each teacher has a set of preferences that should be loaded when his lessons start. To that end, an authorization mechanism was developed using a magnetic card reader and smart cards assigned to the teachers. A teacher who enters the classroom swipes the magnetic card against the reader; the system reads the id, logs the teacher and loads his preferences (e.g., the short-term reasoner settings).

VI. Evaluation

As a first step towards the evaluation of the system, an expert-based heuristic evaluation was conducted in order to identify usability problems regarding the Teacher Assistant application. Heuristic evaluation is the most popular usability inspection method and is carried out as a systematic inspection of a user interface design for usability. It is targeted to find usability problems in the design so that they can be attended to as part of an iterative design process. Heuristic evaluation requires a small set of evaluators to examine the interface and judge its compliance with a set of recognized usability principles. The optimal approach, according to Nielsen [51], is to involve three to five evaluators, since larger numbers do not provide much additional information. An observer notes down the issues and creates an aggregated list, which is delivered at the end to the evaluators in order to provide severity ratings on each issue.

Four usability experts took part in the evaluation of AmI-RIA and identified 22 issues, out of which 11 were marked as severe (rated above 2.5 on a 0-4 scale) and the other 11 as minor ones (rated bellow 2.5). The identified usability errors were related mostly to the flexibility in access to the several components (e.g., the attendance access button) and the perceived user friendliness when operated on touch-enabled devices (e.g., the sidebar option buttons were difficult to press due to their size and were not identified as toggle buttons, Figure 13). Additionally, some issues were identified regarding the aesthetic design and accessibility of the user-interface such as the insufficient color contrast between the main visual components (e.g., the main menu buttons and the footer's information). The released prototype of AmI-RIA effectively addresses all the identified errors.



(a) Initial design (b) Adjusted design based on feedback

Figure 13: GUI adjustment based on the feedback of the experts

VII. Deployment challenges & limitations

The AmI-RIA system builds upon a Smart Classroom environment that provides the infrastructure required to assist the educational process from the students' perspective, and at the same time provides the monitoring facilities that enable the teacher's assistance. The prototype deployment of AmI-RIA[6] in the envisioned Smart Classroom at FORTH-ICS achieved the coordination of the several components (backbone infrastructure, augmented desk, interactive learning environment, smart board, etc.) and proved the feasibility of a real-world installation.

However, due to the monitoring nature of the system a few behavioral, legal and technological challenges arise that need to be addressed prior to deployment in real classrooms. The most interesting and challenging of them is about the willingness of the students to be constantly monitored, even though monitoring is limited to educational activities and does not involve personal ones. Students may grow the feeling of being continuously evaluated by their supervisor, a situation that may cause alterations in their behavior. Towards this, a further study is under work in order to add feedback from the monitoring process to the student's desk so that the students would demystify the procedure. In addition to this, providing some real-time feedback to students will assist them improve their progress by filling in-time any learning gaps that may occur [52], thus motivating students to support the data collection from their desks.

In addition to the students' willingness to participate in such data collecting environments, legal issues need to be considered as well. As the legislation does not cover such scenarios yet, it must be made clear who (if any) will have to give the permission for the participation of each student. Schools should work on a common policy towards student monitoring and decide whether parental permission will be mandatory.

AmI-RIA is aiming to support the average-size classroom, that is around 22 students per each according to Eurostat

[6]AmI-RIA prototype: http://bit.ly/1iuJIFd (Online: 5/2014)

TABLE I: MAXIMUM NUMBER OF SIMULTANEOUSLY VISIBLE VIRTUAL DESKS IN A SINGLE PAGE PER SCREEN RESOLUTION

Screen resolution	Maximum supported desks
1024 X 768 (XGA)	25
1280 X 1024 (SXGA)	49
1400 X 1050 (SXGA+)	56
1600 X 1200 (UXGA)	72

statistics [53], and even larger ones, up to 72 students (or more). The number of students supported is limited by the number of student cards that can be displayed in the classroom view of the Teacher Assistant and is analogous to the screen resolution of the teacher's computer (Table I). However, the system can be used best at a reasonable amount of students (20 - 30) due to the visual load on the Teacher Assistant frontend. Ways of overcoming this limitation are still under study (e.g., paging of student cards).

VIII. CONCLUSIONS

This article has presented AmI-RIA, a real-time system that assists teachers in the context of an intelligent classroom by exploiting the available ambient technology from their perspective. The proposed system monitors the students' activities in an unobtrusive way and generates valuable insights in order to assist teachers keep track of the classroom's progress and performance. Thereby, the teacher is supplied with the needed information to decide when and how to intervene providing help or adapt the teaching strategy.

For that to be achieved, the Desk Monitor agents of AmI-RIA collects all data generated by students' interactions during the educational process and store them in a semantic way, along with their semantic taxonomies. A knowledge extraction mechanism is then used to produce inferences over the data in order to identify potential weaknesses and pitfalls that need to be addressed.

On the teacher's side, the Teacher Assistant application has been implemented to provide a real-time classroom visualizer. Its rich, yet intuitive user-interface, delivers to the teacher all the information required to control the classroom effectively. Furthermore, a Statistics component is introduced, achieving to replace standard reporting of students' progress. Through that component the teacher is able to review and provide grading to students and their parents, but also compare performance across lessons or topics of a lesson, in order to oversee and address learning gaps. Finally, a set of tools have been developed to enhance typical procedures that can be found in conventional classrooms, such as the classroom attendance record, tasks assignment, etc.

IX. FUTURE WORK

The next step of this work will be to conduct a full-scale user-based evaluation in a real classroom. The evaluation is planned to include 20 different teachers and their students [54], where typical classroom activities will be observed to: (i) assess whether AmI-RIA recognizes problems successfully, and (ii) determine how instructors use the system to identify problems and provide assistance. The evaluation's findings are foreseen to extend the currently implemented rule set and improve the user interface of the teacher's frontend application in terms of usability.

Additionally, some relevant topics are being investigated for future upgrades. A significant addition to the system would be to make the students' desks aware about the knowledge generated from the collected data during the reasoning process. This way, the students will get real-time insights about their progress during the various learning activities and this will help them familiarize with the data collection procedure. The feedback provided could be used by the students to adjust their activities accordingly, while communication between the teacher and the students could be also enhanced. For instance, a valuable feature would be to enable the teacher reward some students for achieving high scores on a task or provide extra material to those who had problems in a topic.

Another important extension of the system would be the development of a graphical tool that will facilitate the fast and simple modification of the reasoning rules used to identify students' problematic states. This tool will offer a friendly frontend enabling teachers to manage rules by combining condition facts and defining the desired knowledge extraction. Ideally, entities, properties and values from the data-model will be presented as graphical elements that can be dragged and dropped, building this way the new rules.

Finally, another promising extension of the system would be to develop the infrastructure that will enable the aggregation of the information originating from multiple classrooms. This tool could be used by the school administration in order to keep track of all the students and classrooms. Relevant changes in the schools policies about grading reports may also enable the replacement of conventional reports by more detailed graphical ones, as generated by AmI-RIA.

X. ACKNOWLEDGMENTS

This work is supported by the FORTH-ICS internal RTD Programme 'Ambient Intelligence and Smart Environments'.

REFERENCES

[1] G. Mathioudakis, A. Leonidis, M. Korozi, G. Margetis, S. Ntoa, M. Antona, and C. Stephanidis, "Ami-ria: Real-time teacher assistance tool for an ambient intelligence classroom," in *eLmL 2013, The Fifth International Conference on Mobile, Hybrid, and On-line Learning*, pp. 37–42, 2013.

[2] J. C. Augusto and P. McCullagh, "Ambient intelligence: Concepts and applications," *Computer Science and Information Systems/ComSIS*, vol. 4, no. 1, pp. 1–26, 2007.

[3] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.

[4] J. C. Augusto and C. D. Nugent, *Designing smart homes: the role of artificial intelligence*, vol. 4008. Springer, 2006.

[5] M. Antona, A. Leonidis, G. Margetis, M. Korozi, S. Ntoa, and C. Stephanidis, "A student-centric intelligent classroom," in *Ambient Intelligence*, pp. 248–252, Springer, 2011.

[6] R. A. Ramadan, H. Hagras, M. Nawito, A. E. Faham, and B. Eldesouky, "The intelligent classroom: towards an educational ambient intelligence testbed," in *Intelligent Environments (IE), 2010 Sixth International Conference on*, pp. 344–349, IEEE, 2010.

[7] M. Koutraki, V. Efthymiou, and G. Antoniou, "S-creta: Smart classroom real-time assistance," in *Ambient Intelligence-Software and Applications*, pp. 67–74, Springer, 2012.

[8] G. D. Abowd, C. G. Atkeson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney, and M. Tani, "Teaching and learning as multimedia authoring: the classroom 2000 project," in *Proceedings of the fourth ACM international conference on Multimedia*, pp. 187–198, ACM, 1997.

[9] Y. Shi, W. Xie, G. Xu, R. Shi, E. Chen, Y. Mao, and F. Liu, "The smart classroom: merging technologies for seamless tele-education," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 47–55, 2003.

[10] G. Margetis, A. Leonidis, M. Antona, and C. Stephanidis, "Towards ambient intelligence in the classroom," in *Universal Access in Human-Computer Interaction. Applications and Services*, pp. 577–586, Springer, 2011.

[11] L.-K. Soh, N. Khandaker, and H. Jiang, "I-minds: a multiagent system for intelligent computer-supported collaborative learning and classroom management," *International Journal of Artificial Intelligence in Education*, vol. 18, no. 2, pp. 119–151, 2008.

[12] H. McLellan, *Situated learnng perspectives*. Educational Technology, 1996.

[13] C. Murphy, G. Kaiser, K. Loveland, and S. Hasan, "Retina: helping students and instructors based on observed programming activities," in *ACM SIGCSE Bulletin*, vol. 41, pp. 178–182, ACM, 2009.

[14] I. Jormanainen, Y. Zhang, E. Sutinen, *et al.*, "Agency architecture for teacher intervention in robotics classes," in *Advanced Learning Technologies, 2006. Sixth International Conference on*, pp. 142–143, IEEE, 2006.

[15] M. M. T. Rodrigo and R. S. Baker, "Coarse-grained detection of student frustration in an introductory programming course," in *Proceedings of the fifth international workshop on Computing education research workshop*, pp. 75–80, ACM, 2009.

[16] M. Ahmadzadeh, D. Elliman, and C. Higgins, "An analysis of patterns of debugging among novice computer science students," in *ACM SIGCSE Bulletin*, vol. 37, pp. 84–88, ACM, 2005.

[17] M. C. Jadud, "Methods and tools for exploring novice compilation behaviour," in *Proceedings of the second international workshop on Computing education research*, pp. 73–84, ACM, 2006.

[18] E. S. Tabanao, M. M. T. Rodrigo, and M. C. Jadud, "Predicting at-risk novice java programmers through the analysis of online protocols," in *Proceedings of the seventh international workshop on Computing education research*, pp. 85–92, ACM, 2011.

[19] J. Helminen, P. Ihantola, and V. Karavirta, "Recording and analyzing in-browser programming sessions," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, pp. 13–22, ACM, 2013.

[20] D. Pearce-Lazard, A. Poulovassilis, and E. Geraniou, "The design of teacher assistance tools in an exploratory learning environment for mathematics generalisation," in *Sustaining TEL: From innovation to learning and practice*, pp. 260–275, Springer, 2010.

[21] L.-K. Soh, N. Khandaker, X. Liu, and H. Jiang, "A computer-supported cooperative learning system with multiagent intelligence," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 1556–1563, ACM, 2006.

[22] L.-K. Soh, X. Liu, X. Zhang, J. Al-Jaroodi, H. Jiang, and P. Vemuri, "I-minds: an agent-oriented information system for applications in education," in *Agent-Oriented Information Systems*, pp. 16–31, Springer, 2004.

[23] L. France, J.-M. Heraud, J.-C. Marty, T. Carron, and J. Heili, "Monitoring virtual classroom: Visualization techniques to observe student activities in an e-learning system," in *Advanced Learning Technologies, 2006. Sixth International Conference on*, pp. 716–720, IEEE, 2006.

[24] L. Kepka, J.-M. Heraud, L. France, J.-C. Marty, and T. Carron, "Activity visualization and regulation in a virtual classroom," in *Proceedings of the 10th IASTED International Conference on Computers and Advanced Technology in Education*, pp. 507–510, ACTA Press, 2007.

[25] H. Chernoff, "The use of faces to represent points in k-dimensional space graphically," *Journal of the American Statistical Association*, vol. 68, no. 342, pp. 361–368, 1973.

[26] R. Mazza and V. Dimitrova, "Generation of graphical representations of student tracking data in course management systems," in *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pp. 253–258, IEEE, 2005.

[27] R. Mazza and V. Dimitrova, "Coursevis: A graphical student monitoring tool for supporting instructors in web-based distance courses," *International Journal of Human-Computer Studies*, vol. 65, no. 2, pp. 125–139, 2007.

[28] T. Chronopoulos and I. Hatzilygeroudis, "An intelligent system for monitoring and supervising lessons in lams," in *Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on*, pp. 46–53, IEEE, 2010.

[29] S. Kalyuga, "Rapid cognitive assessment of learners' knowledge structures," *Learning and Instruction*, vol. 16, no. 1, pp. 1–11, 2006.

[30] A. Leonidis, G. Margetis, M. Antona, and C. Stephanidis, "Classmate: Enabling ambient intelligence in the classroom," *World Academy of Science, Engineering and Technology*, vol. 66, pp. 594–598, 2010.

[31] M. Antona, G. Margetis, S. Ntoa, A. Leonidis, M. Korozi, G. Paparoulis, and C. Stephanidis, "Ambient intelligence in the classroom: an augmented school desk," in *Proceedings of the 2010 AHFE International Conference (3rd International Conference on Applied Human Factors and Ergonomics), Miami, Florida, USA*, pp. 17–20, 2010.

[32] G. Margetis, X. Zabulis, P. Koutlemanis, M. Antona, and C. Stephanidis, "Augmented interaction with physical books in an ambient intelligence learning environment," *Multimedia Tools and Applications*, pp. 1–23, 2012.

[33] Y. Georgalis, D. Grammenos, and C. Stephanidis, "Middleware for ambient intelligence environments: Reviewing requirements and communication technologies," in *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, pp. 168–177, Springer, 2009.

[34] M. Korozi, S. Ntoa, M. Antona, A. Leonidis, and C. Stephanidis, "Towards building pervasive uis for the intelligent classroom: the pupil approach," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 279–286, ACM, 2012.

[35] R. Krummenacher and T. Strang, "Ontology-based context modeling," in *Proceedings Third Workshop on Context-Aware Proactive Systems (CAPS 2007)(June 2007)*, 2007.

[36] D. Brickley and L. Miller, "Foaf vocabulary specification 0.99," tech. rep., 2014. http://xmlns.com/foaf/spec/.

[37] Dublin Core Metadata Initiative (DCMI), "Dublin core metadata element set, version 1.1," tech. rep., Dublin Core Metadata Initiative (DCMI), 2012. http://dublincore.org/documents/dces/.

[38] D. Brickley and R. Guha, "Rdf schema 1.1," recommendation, W3C, February 2014. http://www.w3.org/TR/rdf-schema/.

[39] T. Tudorache, J. Vendetti, and N. F. Noy, "Web-protege: A lightweight owl ontology editor for the web.," in *OWLED*, vol. 432, 2008.

[40] W3C SPARQL Working Group, "Sparql 1.1 overview," recommendation, W3C, March 2013. http://www.w3.org/TR/sparql11-overview/.

[41] J. Tauberer, "Semweb: A .net library for rdf and the semantic web." http://razor.occams.info/code/semweb/semweb-current/doc/, 2010. [Online; accessed May-2014].

[42] J. D. Roo, "Euler yet another proof engine." http://eulersharp.sourceforge.net/, 2014. [Online; accessed May-2014].

[43] T. Berners-Lee and D. Connolly, "Notation3 (n3): A readable rdf syntax," tech. rep., W3C, March 2011. http://www.w3.org/TeamSubmission/n3/.

[44] J. Carroll, "A model of school learning," *The Teachers College Record*, vol. 64, no. 8, pp. 723–723, 1963.

[45] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner, "Off-task behavior in the cognitive tutor classroom: when students game the system," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 383–390, ACM, 2004.

[46] M. Cocea, A. Hershkovitz, and R. Baker, "The impact of off-task and gaming behaviors on learning: immediate or aggregate?," 2009.

[47] J. E. Beck, "Using response times to model student disengagement," in *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, pp. 13–20, 2004.

[48] Learning Technology Standards Committee of the IEEE, "Draft standard for learning technology - learning object metadata," tech. rep., IEEE Standards Department, New York, July 2002.

[49] O. Medvedeva, A. M. de Carvalho, R. S. Baker, and R. S. Crowley, "A classifier to detect student gamingof a medical education system,"

[50] D. A. Norman, *The design of everyday things*. [New York]: Basic Books, 2002.

[51] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 249–256, ACM, 1990.

[52] D. Curtis and M. Lawson, "Collaborative online learning: An exploratory case study," in *International Conference of Merdsa, Melbourne*, vol. 44, 1999.

[53] Eurostat, "Pupil/student - teacher ratio and average class size (isced 1-3)." http://bit.ly/1aR4jqn, 2014. [Online; accessed May-2014].

[54] J. Nielsen, "Quantitative studies: How many users to test?." http://www.nngroup.com/articles/quantitative-studies-how-many-users/, 2006.