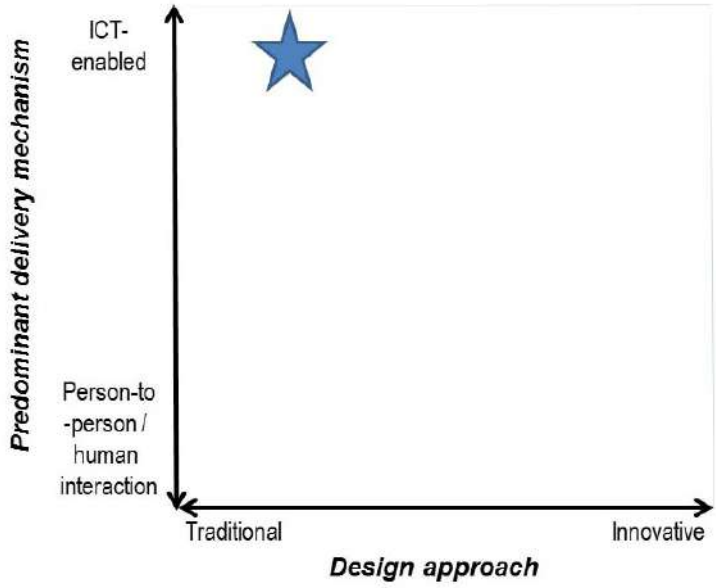
Chapter thirteen

ICT Tools and Approaches to Support and Enhance Case Based Learning

Stefanos Petsios, Petros Karvelis and Chrysostomos Stylios



Introduction

Information and Communication Technology (ICT) tools are continuously and increasingly used for learning and educational procedures. Many educational platforms have been proposed to support learning and today these typically take advantage of wireless

communications and the Internet to enable online learning. Although such ICT tools and participation technologies could enable these educational platforms to be used for Case Based Learning (CBL) relatively few have been specifically developed for this purpose. This chapter considers the extent to which pedagogical needs are met by the existing educational platforms and their tools.

Most features of educational platforms give the opportunity to create online student/teacher communities whose participants can link variously via PCs, mobile phones, tablets, web TVs etc. The generally accepted pedagogical model today offers a combination of off-line teaching methods and real time participation and interaction with the e-students. In this chapter we will describe the most representative educational platforms that support CBL. We will categorise them according to their features and their capabilities to fulfill pedagogical requirements and instructional strategies.

Pedagogical Model Requirements for Case Based Learning

A case based pedagogical model incorporates the possibility of working with more than one problem at a time. In this approach the students have to combine, choose and prioritise different disciplines and problems. CBL promotes the development of a collaborative, personal or team-based teaching approach suitable for any education field.

Case based learning is similar to problem-based learning but it is also considered as an iterative procedure covering problem orientation, problem solving and innovation. Important characteristics of CBL include hypothesis generation and the integration of learning activities. In the case based pedagogical model every case aims to teach specific attributes. In the field of business studies these will typically include organisation, risk treatment, risk management, and communication – in short, management that is faster, cheaper, better and easier. However, case based learning is not limited solely to working with cases (McNair, 1954; Bolt B, 1998; Flynn & Klein,

2001; Yadav, 2007; Zbylut, 2007; Bradley, 2004; Crawa, 2006; Lombardi, 2007; Iacono, 2011).

Instructional Strategies for Case Based Learning

Advances in ICT are having a major impact on the way people do business, access and share information, and create and transfer knowledge. Teaching and learning strategies and their procedures have to be updated to prepare students to cope with these new situations. Students need to be able to pose questions, seek and find appropriate resources for answering these questions, collaborate and co-work with others, and then to communicate and present their solutions effectively to others.

Duch *et al.* (2001) offer extensive descriptions of the desirable skills that should flow from a problem based learning strategy. In particular these include the ability to:

- think critically and be able to analyse and solve complex, real-world problems
- find, evaluate, and provide appropriate learning resources
- work cooperatively in teams or small groups
- demonstrate versatile and effective communication skills, both verbal and written
- use content knowledge and intellectual skills acquired during higher education in order to become continual learners.

Consideration of Alternative Instructional Strategies

Various instructional strategies have evolved which can complement or offer an alternative to CBL. The following examples set out key aspects.

Problem based learning (Srinivasan, 2007) is both a teaching method and an alternative to the traditional curriculum. It includes

carefully designed problems that challenge students to use problem solving techniques, self-directed learning strategies, team participation skills and discipline-specific knowledge. Problem based learning can be very effective in the learning process concerned with the natural sciences.

Challenge based learning (Challenge Based Learning Organisation) has its roots in the problem based learning approach. There is a redefined approach which focuses on increasing student engagement, especially for students most at risk of dropping out. In challenge based learning a collaborative learning experience is performed where teachers and students work together to learn about compelling issues, to propose solutions to real cases and to take action. This approach requires students to reflect on their learning and the impact of their actions, and then publish their solutions for the benefit of learners anywhere in the world.

The *Role play and debates* approach is usually a real-time teaching interaction. Fortin (2012) promotes it as one of the best ways to educate people – and, in particular, students who are potential future teachers. The real time interaction accelerates learning of the skills of speed and real time adaptation. Achieving real time online teaching and learning through synchronous web-based conference platforms is an essential component of any e-education platform – especially when there are high levels of demand but limited hardware resources and network infrastructure. When the online teaching is delivered asynchronously it often deploys text-based chat or forum platforms to enable debates about matters of relevance to the course.

Virtual Environments and Simulation of real cases can be particularly effective when training students to deal with 'close to reality' situations. This is a novel and emerging approach with many possible expansions and features that may be escalated to provide virtual worlds for educational and skills training purposes. The technological trends in portable devices and upgraded metropolitan infrastructures enable the use of augmented reality tools that simulate the real world.

A*nalysis and reflection* are capabilities that are prioritised in many

Higher Education institutions. However, the scope of the analysis stage is often very wide and under-specified and this can make it very hard for students to discern and address the real case problems. To remedy this, tutors may have to gather the students' reflections and motivate them to perform new and better analysis.

Long term projects can be a highly effective teaching strategy for team building and for developing skills to work collaboratively. Learning and working in groups involves shared and learned values, resources and ways of implementing tasks. Teams learn how to succeed by combining these factors. The effectiveness of a team – and of its individual members – will depend on their ability to respect differences within the team.

ICT Tools for Case Based Learning

There are many learning platforms offering a variety of features and specifications, The main criteria for selecting the platforms are that they should be used by a large number of users and should to be actively supported.

There is a recent trend on web-based education platforms towards having an up to date web browser as a minimum client requirement. In the following section we envisage the set of features that would make an ideal platform focusing on Case Based Learning.

A Review of Representative E-Learning Platforms

Numerous ICT tools and platforms are available for tutoring that their vendors claim have features for case based learning. Table 1 (below) presents in alphabetical order a representative sample of the currently available platforms that are providing CBL tools. The key characteristics of these are then summarised and evaluated. Table 2 then compares the reviewed platforms on the basis of the features and functionality offered.

Platform	Web Page
ATutor	http://www.atutor.ca
Dokeos	http://www.dokeos.com
dotLRN	http://www.dotlrn.org
ILIAS	http://www.ilias.de
LON-CAPA	http://www.lon-capa.org
Moodle	http://www.moodle.org
OpenACS	http://www.openacs.org
Sakai	http://www.sakaiproject.org
Docebo	http://www.docebo.com
Claroline	http://www.claronie.net
SLED	http://www.secondlife.com
Shareville	http://www.shareville.bcu.ac.uk

Table 1: Platforms provided features for Case Base Learning

ATutor is a standards-compliant, Web-based Learning Content Management System (LCMS) developed by the Adaptive Technology Resource Centre of the University of Toronto. It is open-source software compliant with the GNU Project Standards. This means that course content created in ATutor and other compliant packages can be exported/imported from one to the other.

Dokeos (Scalise & Gifford, 2010) is a platform for distance learning (ie an e-learning platform) which is intuitive and easy to use mainly by trainers, learners and continuing education auditors. Dokeos allows trainers to focus on creating scenarios and content by freeing them from any technical aspect. It also provides collaborative tools: videoconferencing, forums, blogs, and wiki etc. Dokeos includes four main components, namely; AUTHOR to build e-learning content, LMS to handle interaction with learners, SHOP to sell a course catalog, and EVALUATE for assessment and certification.

.LRN (pronounced "dot learn") is actually a global community of educators, designers, and software developers who partner together to drive educational innovation. This software is open source and this enables organisations to invest in people and curriculum development instead of expensive licensing and support fees.

ILIAS is an open source web-based learning management system (LMS). It supports learning content management and tools for collaboration, communication, evaluation and assessment. The software is published under the GNU General Public License and can be run on any server that supports PHP and MySQL.

LON-CAPA (Learning Online Network with Computer-Assisted Personalized Approach) is an e-learning platform which possesses the standard features of many learning platforms (user roles, calendar, e-mail, chat rooms, blogs, resource construction, test grading, etc.). The main advance of traditional e-learning platforms is that the web servers can communicate with each other. Consequently, the term LON-CAPA also refers to the LON-CAPA network, i.e. the entire set of LON-CAPA web servers and the specific implementation of an internet protocol that connects these web servers.

Moodle (acronym for Modular Object-Oriented Dynamic Learning Environment) is a free software e-learning platform, also known as a Learning Management System, or Virtual Learning Environment (VLE).

Moodle has several features considered typical of an e-learning platform, plus some original innovations (like its filtering system). Moodle is very similar to a learning management system. Moodle is widely used in a range of environments such as education, training and development, and business settings.

OpenACS (The Open Architecture Community System) is an open-source web application framework.

The Open Architecture Community System provides:

- A set of applications that are used to deploy web sites and are strong on collaboration. Some of the applications are Workflow, CMS, and Messaging, Bug/Issue tracker, e-commerce, blogger, chat and forums.
- A sophisticated application development toolkit that provides an extensive set of APIs and services to enable quick development of new applications. Features include sophisticated permissioning, full internationalisation, Ajax, form builder, object model, automated testing, sub-sites and a powerful package manager.

Sakai is a community of academic institutions, commercial organisations and individuals who work together to develop a common Collaboration and Learning Environment (CLE). The Sakai CLE is a free, community source, educational software platform distributed under the Educational Community License (a type of open source license). The Sakai CLE is used for teaching, research and collaboration. Systems of this type are also known as Course Management Systems (CMS), Learning Management Systems (LMS), or Virtual Learning Environments (VLE).

Docebo. The Docebo suite is a completely free content management (CMS) and e-learning (LMS) platform released under Open Source license.

Claroline is a collaborative eLearning and eWorking platform (Learning Management System) released under the GPL Open Source license. It is used in hundreds of organisations worldwide ranging from universities to schools and from companies to associations to create and administer courses and collaboration spaces over the web. Claroline is used in more than 100 countries and is available in 35 languages.

Second Life (SLED) is an immersive, online, simulated environment,

with 3-D graphics that allow users to interact by means of an avatar – a virtual embodiment of a person (or oneself) that mimics real-life interactions. SLED has a user base of thousands of registered avatars. These avatars can be used for many different purposes such as gaming, social networking, marketing and commerce and real world business. The software incorporates a three-dimensional modeling tool based on simple geometric shapes that allows avatars to build virtual objects.

Shareville is a backronym for Shareable, Holistic Assets and Resources, Existing in a Virtual Interactive Lifelong Learning Environment (Staley & Faniglione, 2010; Lowe *et al.*, this volume). The hardware specification required to run the environment is minimal and, being web-based, it works on a variety of computer platforms. Shareville provides an approximation of the socially and ethnically diverse city of Birmingham, England, and many of the local areas and landmarks within the city are parodied in the names used within Shareville.

Figure 1: The Shareville Map.

Navigation within the Shareville environment is possible through 'point and click'. At the top of each page there is a breadcrumb trail

to improve accessibility, alongside quick navigation to the locations. There is no 'full screen' mode or fully immersive environment. The perspective of the student is always 'first person', so there is no requirement for an avataristic representation on screen. Whilst not a JISC-sponsored project, the Joint Information Systems Committee has demonstrated interest in Shareville, including presentations at online conferences (Staley, Mackenzie, Hetherington & Faniglione, 2009).

Table 2 presents a comparison of the reviewed platforms based on their provision (or not) of features or attributes which we consider to be essential.

	ATutor	Dokeos	dotLRN	ILIAS	LON CAPA	Moodle	Open ACS	Sakai	Docebo	Claroline	SLED	Shareville
Available free of charge	✓				✓	✓	✓	✓		✓		
Platform		✓			✓	✓	✓	✓	✓	✓		
Interface		✓			✓	✓	✓	✓	✓	✓	✓	✓
User Account		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Email	✓	✓	✓	✓		✓	✓	✓		✓		
GradeBook	✓	✓	✓	✓	✓	✓	✓	✓		✓		
FAQs	✓		✓	✓		✓	✓	✓	✓	✓		
File Storage		✓	✓			✓	✓	✓		✓		
Forums	✓	✓			✓	✓	✓	✓	✓	✓		
News	✓	✓				✓		✓	✓	✓		
Survey	✓	✓			✓	✓		✓	✓	✓		
application for Android Devices									✓			
application for iOS Devices									✓			

	ATutor	Dokeos	dotLRN	ILIAS	LON CAPA	Moodle	Open ACS	Sakai	Docebo	Claroline	SLED	Shareville
Web TV or Smart TV												
Real time participation and interaction												
problem based learning	✓	✓				✓	✓					✓
challenge based learning	✓	✓				✓	✓		✓			
role play and debates												✓
simulation												✓
analysis and reflection	✓					✓	✓					
group projects	✓					✓	✓					

Table 2: Comparison of the features provided by learning platforms reviewed

E-Learning Platforms which Support Case Based Teaching

A successful e-learning platform has to satisfy certain rules in order to be effective and able to respond to a well-defined scenario. The main requirement of a web case based platform is to support the authoring of cases and to provide and elaborate different types of learning activities based on cases. In our view the essential components of a case based teaching platform are:

- Users: The personal profile of the user.
- Articles: The abstract and the analytical description of the case.

- Keywords: Each case should be described by a number of keywords.
- Forums: A place where the users can communicate, collaborate, chat and ask questions with other users.
- Questionnaires: online tests where questions and multiple choice answers are presented to the student.
- Administration of each case.

Table 3 presents a comparison of the case based features for different e-learning platforms. In order to compare and grade each platform we have introduced the following formula which reflects the value of adoption of the different case based teaching features mentioned above:

$$\text{Grade of Platform} = \frac{\text{number of Case Based Features}}{\text{total number of Case Based Features}} * 100.$$

Platforms	ATutor	Dokeos	dotLRN	ILIAS	LON-CAPA	Moodle	OpenACS	Sakai	Docebo	Claroline	SLED	Shareville
Users	-	+	+	+	+	+	+	+	+	+	+	+
Articles	+	+	+	+	+	+	+	+	+	+	+	+
Keywords	-	-	-	-	+	+	-	+	+	+	-	+
Forums	+	+	-	-	+	+	+	+	+	+	-	-
Questionaires	+	+	+	-	-	+	+	-	+	+	-	+
Administration	+	+	+	+	+	+	+	+	+	+	+	-
Grade	67%	83%	67%	50%	83%	100%	83%	83%	100%	100%	50%	67%

Table 3: Comparison of the reviewed e-learning platforms based on the features provided for case based teaching and learning

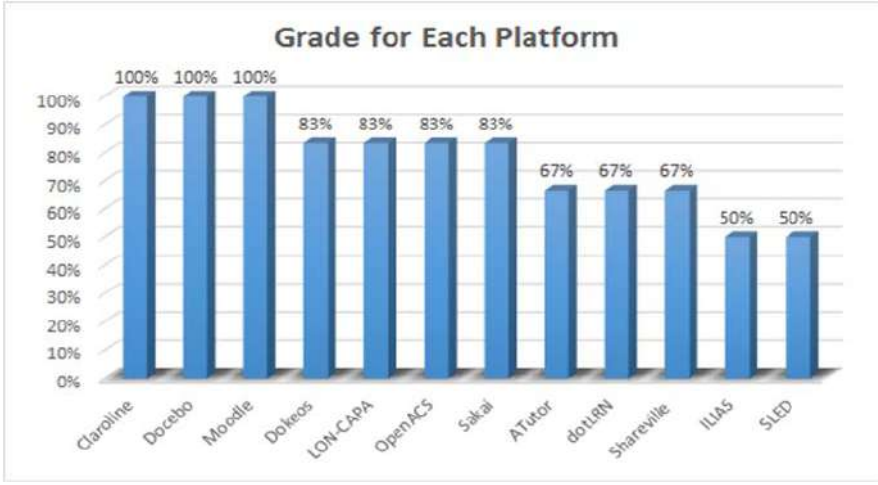


Figure 2: Indicative grading of the reviewed platforms based on the proportion of essential features provided.

Figure 1 indicates that Claronine, Docebo and Moodle can support all the essential features needed for case based teaching. Furthermore two out of the three best platforms Claronine and Moodle are free to use under the General Public License (GPL) license; the Docebo platform is offered as a paid service.

Case Based Learning in Computer Science Teaching

Computer science teaching poses a challenge to case based teaching and learning pedagogies because, unlike the other sciences, computer science deals with problems from a wide spectrum of unrelated disciplines. However, we observe that teaching in computer science is generally still based on traditional teaching approaches. Below, we propose how case based teaching principles could be used to design and conduct two undergraduate computer science courses, namely; Introduction to Programming and Software Engineering.

Example 1: a Procedural Programming Language Course

The learning of code programming is central in computer science and it requires a range of activities to be developed: learn the language features and attributes, outline the program design, and build and comprehend a program. Typically the textbooks for programming courses suggest similar learning methods – starting with declarative knowledge about a particular programming language. However, case based teaching is suitable for application in any programming course. For example, Esteves *et al.* (2010) report a recent attempt where the Second Life environment has been used to teach Programming

Course Description. The course considered here is Introduction to Programming. The course is run during the first year of a four year undergraduate program leading to a Bachelor degree in Computer Engineering at the Technological Educational Institute of Epirus, Greece. The course is taught over a period of 13 weeks; every week involves two hours for theory, two hours for tutoring and two hours of practice in the computer laboratory.

The course covers the basic programming principles and techniques for the C programming language. Procedural programming languages are based on the concept of the unit and scope (the data viewing range of an executable code statement). A procedural program is composed of one or more units or modules, which are either user coded or provided in a code library. Each module is composed of one or more procedures, also called a function, routine, subroutine, or method, depending on the language.

The main topics include Variables, Loops, Conditions, Functions and Files. This course aims to provide students with the necessary knowledge and ability to write their own very basic C applications (Robins *et al.*, 2003).

Teaching Strategy through Cases. Our literature review revealed two quite different proposed structures for a course on Programming;

one by Lin *et al.* (1992), the other by Spooner *et al.* (1997). These differences can be seen in Table 4.

Structure A (Lin *et al.*, 1992)	**Structure B** (Spooner *et al.*, 1997)
1. Programming Problem statement 2. Solution process description 3. Code listing 4. Study questions 5. Test questions	1. Motivation 2. Background 3. Algorithm development 4. New Programming concepts 5. Solution program 6. Discussion 7. Further study

Table 4: Two structures for a Programming course

In our view structure B is the most suitable for case based teaching and learning for the following reasons.

1. Motivation. This section defines the problem and aims to attract the student's interest by providing a simple motivation
2. Background. Several details and necessary information are provided to assist the student to solve the problem. For example, pointers highlight to the student other types of information such as web links and books.
3. Algorithm Designing. A stepwise approach is suggested to encourage good software engineering practices such as functional programming. The user is asked to outline the program that he has designed.
4. New Programming Concepts. This section addresses the question of what new programming concepts are needed to implement the algorithm designed in step 3.

5. Solution Program. An indicative complete program which solves the problem is set out.
6. Discussion. Here a discussion is inaugurated to enable all the students on the course to discuss the provided solution program and the meaning of the results.
7. Further Study. Students are provided with pointers to external links that could help them to explore alternative programming solutions.

Example 2: a Software Engineering Course

Our Software Engineering course is usually taught during the 6th or 7th semester; it aims to teach students to develop integrated IT systems. We find that a case based learning approach in most suitable for this course because it is ideal for use by small groups of technologically mature and motivated students. A dedicated faculty tutor will teach them basic software engineering concepts in the context of real software engineering cases. The proposed process follows the following basic steps:

- A real application software engineering problem is presented to the students who are arranged into small groups. They organise their ideas to deal with the given problem by using their existing knowledge and then attempt to define the broad nature of the problem.
- Through discussion, students pose questions to each other in order to delineate aspects of the problem that they do not understand. The answers to the questions are organised to create a Software Requirement Analysis. This analysis is evaluated by the tutor who poses new questions to the team and continually encourages the students to define what they know about the problem and, more importantly, what they don't know. The finalised requirements analysis exposes the analysis of the real problem.

- Then the students work in groups to update the software requirement analysis and to describe the final functional and non-functional requirements of the system. In each stage of the analysis the students develop a corresponding study (such as a feasibility study and a study of anticipated costs, etc.) that the tutor has to evaluate. This is done not by error correction but by posing relevant questions or dilemmas so that students will update their reports and minimise the gap with the real case scenario.
- The next stage of the case based learning approach requires the group to start the modular software design and software architecture. Each member initiates a detailed analysis of a system module and tests all the modules of the other group members. In this phase, students work on their own but they also have to collaborate with the other team members. The modular approach helps students to improve their skills in interfacing, reusability, cost evaluation, implementation cost, and effort estimation. At this stage we do not focus the students on the software implementation of the project but on creating a proper full software design report. This will use software illustration procedures such as use cases diagrams, activity and state diagrams, collaboration diagrams and etc. These diagrams would be used by any qualified software developer in order to create the integrated software which solves the real problem.
- Finally, the students compare their outputs with the indicated real problem output and reflect on what they have learned, integrating their new knowledge in to the context of the problem. Students are also encouraged to summarise their knowledge and connect new software engineering problems to old ones. They continue to define new learning approaches as they progress through the problem. By now students will understand that Software Engineering is an ongoing process with new methodologies and techniques

even though the principles regarding the system requirement description remain the same.

In the Computer Engineering department of the Technological Educational Institute of Epirus, Greece, the Software Engineering course incorporates the case based learning process described above. An e-learning approach based on the Moodle platform is now being developed to coordinate the weekly assignments, resources and case based learning material. Moreover, we are introducing the use of the kaltura add-on (kaltura.org) as a video library of useful in class recordings and the bigbluebutton (bigbluebutton.org) add-on for group and tutor sessions to evaluate the progress on the real software engineering case.

Conclusions

This chapter has described a representative sample of available e-learning platforms and set out the features each has that would make them suitable for developing case based teaching. These analyses indicate that the Moodle platform is the most suitable one because it offers all the features needed. Furthermore, it is a free platform that is actively supported by a world-wide community of user/developers and provides a large number of features. However, no single platform can be the perfect one for case based teaching and learning. Designers and developers should realise that a new case based teaching platform will not always stand alone; it has to be able to work in conjunction with other ICT platforms such as kaltura for on line video storage and big blue button for webinars.

Most importantly, tutors have to realise that case based teaching platforms require their support during the design and development stage so they will be able to build efficient databases of case examples. In addition they will need ICT tools which incorporate mechanisms to produce efficiency metrics at the end of each case based learning session. This will enable the tutor and developer to evaluate and update the session.

The improved functionality offered by available case based teaching approaches is leading to increasing take up by educators. Furthermore, almost every significant platform provides regular updates and new useful features.

We recommend that the designers of such systems should enable more and more cross-platform operations and inter-operability. This could be achieved either with an external add-on or with an embedded module. In our view this would significantly increase the value and effectiveness of case based learning.

The user interfaces for all the tools we have examined in this chapter are well designed web platforms providing the user with excellent browsing via a personal computer. However the same cannot be said for portable devices such as the tablets and smartphones that are becoming increasingly popular. We exhort interface designers to focus on creating modern, responsive web interfaces and bespoke applications for iOS or Android-based devices. This is because a major success factor of e-platforms is the ability to learn wherever and whenever you want – including the comfort of the TV room. Accordingly tutors should press the platform designers to create intuitive interfaces for their platforms which are aligned with WebTV and Smart TV protocols. Such features would also prove extremely useful in the classroom because the tutor, using an installed Smart TV could navigates to a movie or video or web source, discuss this with his class, and then encourage the students to use their smart phones to search for relevant information that will strengthen their knowledge. The addition of linkages to a cloud computing shared disc space could further interest and engage students.

Case based teaching and learning platforms need to be capable of inter-operating with third party platforms. In our observation existing virtual worlds and educational tools for simulation lack the support of add-ons for creating easy and fast new scenarios. The solution will be for developers to create a universal prototyping and modelling language for creating real case scenarios in a formal and machine readable format. Such a protocol, when accompanied by the necessary ICT tools, will promote the reusability and modular construction of

case based learning materials and provide a better quality of features and services to the students and tutors.

About the Authors

Stefanos Petsios is a Researcher at the Laboratory of Knowledge & Intelligent Computing of the Department of Computer Engineering at the Technological Educational Institute of Epirus, Greece. He can be contacted at this email: stefanos@kic.teiep.gr

Petros Karvelis is a Postdoctoral Researcher at the Laboratory of Knowledge & Intelligent Computing of the Department of Computer Engineering at the Technological Educational Institute of Epirus, Greece. He can be contacted at this email: pkarvelis@gmail.com

Chrysostomos Stylios is Associate Professor in the Department of Computer Engineering at the Technological Educational Institute of Epirus, Greece. He can be contacted at this email: stylios@teiep.gr