Artificial Intelligence in Medicine (2006) **38**, 67—78

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Web-based adaptive training simulator system for cardiac life support

Cristóbal Romero [a,*], Sebastián Ventura [a], Eva L. Gibaja [a], Cesar Hervás [a], Francisco Romero [b]

[a] *Department of Computer Sciences, University of Cordoba, 14071 Campus de Rabanales, Cordoba, Spain*
[b] *Public Company of Health Emergencies (EPES) 061, 23002 Hospital Dr. Sagaz, Jaen, Spain*

Received 15 February 2005; received in revised form 25 January 2006; accepted 26 January 2006

KEYWORDS
Emergency medicine;
Patient simulation;
Web-based education;
Adaptive intelligent systems

Summary

Objective: We introduce a web-based adaptive training simulator system to exercise cardiopulmonary resuscitation skills. Our purpose is to provide emergency physicians with an additional training tool for cardiac life support clinical cases, by integrating an adaptive learning environment with a web-based case simulator.
Methods and materials: Adaptive systems reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. Our system follows a stage-based learning model with several steps to personalize student learning. First, students learn the theory and content of life support and take computerized tests to evaluate their declarative knowledge of these areas. Second, they practice with clinical case examples and complete an exam at the appropriate level of difficulty to assess their practical knowledge. Finally, they train with additional clinical cases.
Results and conclusion: In order to evaluate the usefulness of the system, we used it in two traditional advanced life support courses at the Jaen Hospital in Spain, as an additional and complementary tool within the course. Results show that the use of adaptation techniques can improve student performance.

1. Introduction

The introduction of computers into medicine is one of the most important events in medical education [1]. Although the expectations have been high, the widespread use of applications has been less than promised. The World Wide Web can play an important role in providing distance-based education. Web-based education or web-based training can be defined as any purposeful application of web technologies to the task of educating a fellow human being [2]. The term web-based training is

* Corresponding author. Tel.: +34 957 218630; fax. +34 957 218630.
E-mail address: cromero@uco.es (C. Romero).

0933-3657/$ — see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2006.01.002

most often used within industry and it involves teaching specific skills, while the terms web-based education and e-learning are more common within universities and involve teaching conceptual knowledge. Traditional web-based educational and training systems are simply a network of static hypertext pages [3]. This leads to orientation and comprehension problems for students since navigation in such courses is completely unrestricted. Adaptive systems for web-based education (ASWE) provide a superior alternative because they adapt to the student or task. They are the result of the joint evolution of intelligent tutoring systems (ITS) [4] and adaptive hypermedia systems (AHS) [3] and they combine the most advantageous features of both, namely, increasing the student's interaction with the educational system and adapting it to the needs of each individual. ITS use knowledge about the domain, the student and the teaching strategies to support flexible individualized learning and tutoring. AHS apply different forms of user models to adapt the content and the links on the hypermedia pages to the user. Many adaptive web-based systems have been developed, targeting various application domains. In educational applications some examples [5] are: ELM-ART, Interbook, KBS-Hyperbook, NetCoach, AHA!, etc. Although most of them are general-purpose educational systems, some are specific to medical education. For example, Medtec [6], is aimed at medical training in anatomy. Other examples of specific artificial intelligence applications for medical education and training are: GUIDON [7], a tutor to identify the most likely causative organism in cases of infectious meningitis and bacteremia; Anatom-Tutor [8], an intelligent anatomy tutoring system; COMET [9], a collaborative ITS for medical problem-based learning; and SlideTutor [10] a model-tracing ITS for teaching microscopic diagnosis. Nevertheless none of these employ an underlying simulator.

Unlike other industries (aviation, nuclear power, etc.) medicine has been slow to develop simulation as a training tool and method to assess performance. Some simulators like CIRCSIM-TUTOR [11], a system to learn about the reflex control of blood pressure; and OncoTCap [12], that provides a cancer modeling laboratory, were developed, but only recently have they come into more widespread use in general medical and specialized education and training. This is partly due to greater awareness of the importance of patient safety and the roles of human performance and work environment in contributing to human error. Simulation based systems have the advantage of providing safe environments in which students can make errors without harming patients. A second advantage of simulation based systems is that they can be used to evaluate competence. The most common application areas for simulations include anaesthesia, radiology, surgery, gastroenterology and cardiology. More advanced simulators have ITS features making the simulation more intelligent. Some examples of medical simulation ITS are: InterSim [13], an interactive simulation based tutor with intelligent assistance for the ear domain; Cardiac Tutor [14], an intelligent simulation-based tutor for advanced cardiac life support; BioSimMER [15], a virtual reality training tool that pits rescue teams against computerized terrorist attacks; and UVIMO [16], a virtual reality training simulation system for medical emergencies.

In general, medical simulators vary in their characteristics and accuracy, and we can distinguish between five main types [17]:

Screen-based simulators are computer instruction programs, which are a part of the medical workplace. The user indicates what his sequence of actions is by selecting them from the interface. They are inexpensive and portable, allow for testing large numbers of users simultaneously, and can automatically evaluate performance and provide customized feedback.

Virtual reality (VR) is a computer technique to generate spaces and objects in three-dimensional representations with multiple sensory feedback cues. VR systems are spread along a continuum of cost versus realism but they require significant computing power.

Training devices simulate real world devices that allow students to acquire the necessary skills for a specific task prior to patient contact.

Realistic simulators are realistic human simulators and in general they include a life-like mannequin which simulates a real patient. They are the most accurate patient simulators but they are very expensive and require dedicated space and trained observers.

Web-based simulators are similar to screen-based simulators but they are designed to be executed remotely over the Internet. Some examples of web-based simulators aimed at emergency medicine are: CPR Simulator [18], a cardiopulmonary resuscitation simulator; and Emergency Simulator [19], an adult and paediatric life support simulator. Nevertheless none of these web-based simulators provide adaptive features.

Tests or quizzes are among the most widely used and well-developed evaluation tools in higher education, there are two main types of web-based testing control algorithms [20]: adaptive and classic. The main advantage of computerized adaptive tests is that each examinee receives different questions and there are usually fewer questions needed than in a classic test [21]. Some examples of adaptive computer tests used in web-based adaptive systems are: SIETTE [22] a system of intelligent evaluation using tests and Test Editor [20] an authoring tool for building adaptive and classic web-based tests in AHA!. We used the Test Editor's algorithms in our system.

In this paper, we describe a system which links computer-based case simulations with ASWE in order to obtain a more intelligent emergency medicine e-learning and e-training system, and to increase confidence in emergency knowledge and skills. The novelty in our approach is the integration of emergency domain knowledge and cognitive skills with adaptation techniques in a web-based case simulation training system. The developed ASWE incorporates the following intelligent components: control of student's learning stages, adaptive content presentation and navigation, adaptive computer testing to evaluate the declarative knowledge, plan-based case simulation system to evaluate procedural knowledge, adaptive selection of difficulty level in advanced cases, recommendation of contents based on student errors and cases, and a rule-based inference system to make pedagogic decisions.

First, we describe the system's architecture and its main components. Second, we detail the implementation of the system. Third, we show the evaluation of this system and the results obtained. Finally, we summarize the main conclusions and plans for future research.

2. System architecture

The system architecture and its main components are shown in Fig. 1. This is a web-based architecture [2] in which all the system's components are stored on a web server and users interact remotely by using a web browser. Components of the system include:

- *User interface*. There are two different user interfaces. The student interface presents personalized information provided by the adaptation model, captures all of the user input and sends it to the adaptation model. The teacher interface lets instructors develop educational materials (tests, cases and contents), and visualize the student's results.
- *Case simulation model*. Contains the procedural knowledge about the resolution plan of each case study. The adaptation model uses this information to control the execution of cases to each particular student.
- *Domain model*. Contains the declarative knowledge about cardiac life support (courses and chapters), educational material (exposition contents,

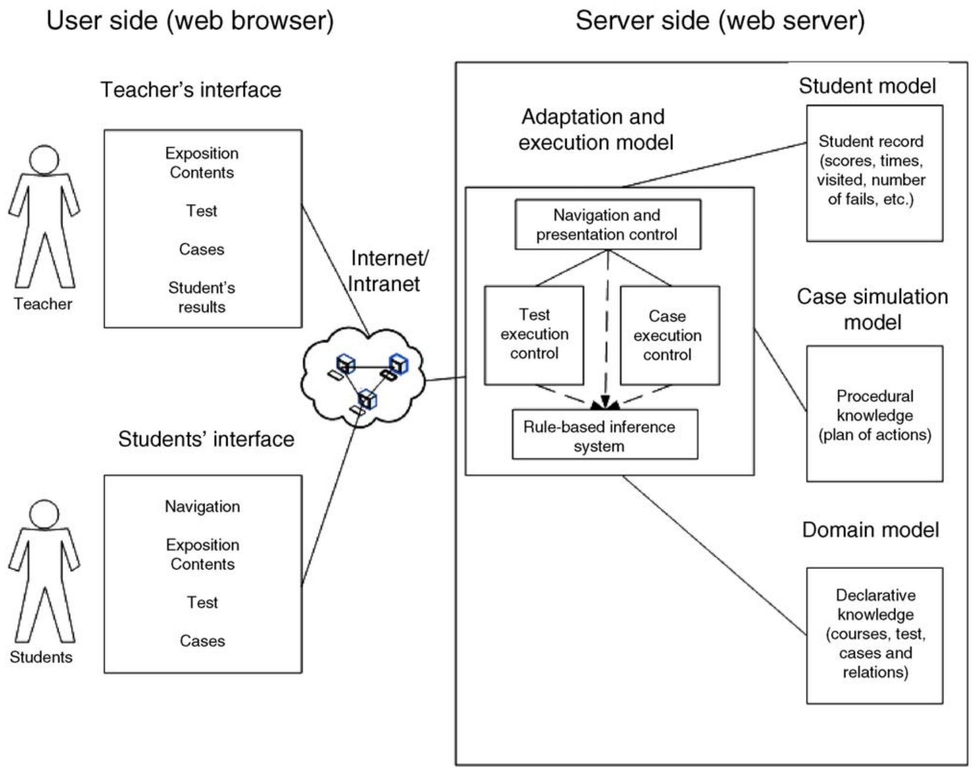


Figure 1 System architecture.

tests and cases) and their relations. The adaptation model uses this information to adapt navigation and presentation to each student.

- *Student model*. Contains a student record for each user with all the interaction and evaluation information. The adaptation and execution model updates this model, and bases pedagogic decisions on its state.
- *Adaptation and execution model*. The engine that controls all the components. It consists of three sub-models (one to control the navigation and the presentation of contents, one to control the execution of tests and one to control the execution of the case simulation). The adaptation and execution system uses a rule-based inference system to make decisions. The adaptation and execution system controls the interface and uses the information from the student model to adapt the domain model and simulation model to each particular student.

2.1. User interface

The system has two different web-based user interfaces (the student interface and the teacher interface).

2.1.1. Student interface

The student interface uses three different types of interaction (browsing, drill and practice, and simulation) [2] by which students interact with the following components of the system:

- *Navigation*. A web interface to guide students through the learning stages, the exposition contents, tests and cases.
- *Exposition contents*. The same web interface with multimedia elements to see and browse the explanatory contents.
- *Test*. An interactive interface to execute computerized tests.
- *Cases*. The most complex interface which students use to interact with simulated cases (Section 2.2).

2.1.2. Teacher interface

Authoring, administrative, and maintenance tasks are performed in the teacher interface. The system provides several authoring tools in order to facilitate the development of exposition contents, tests and cases. Four different tasks are supported:

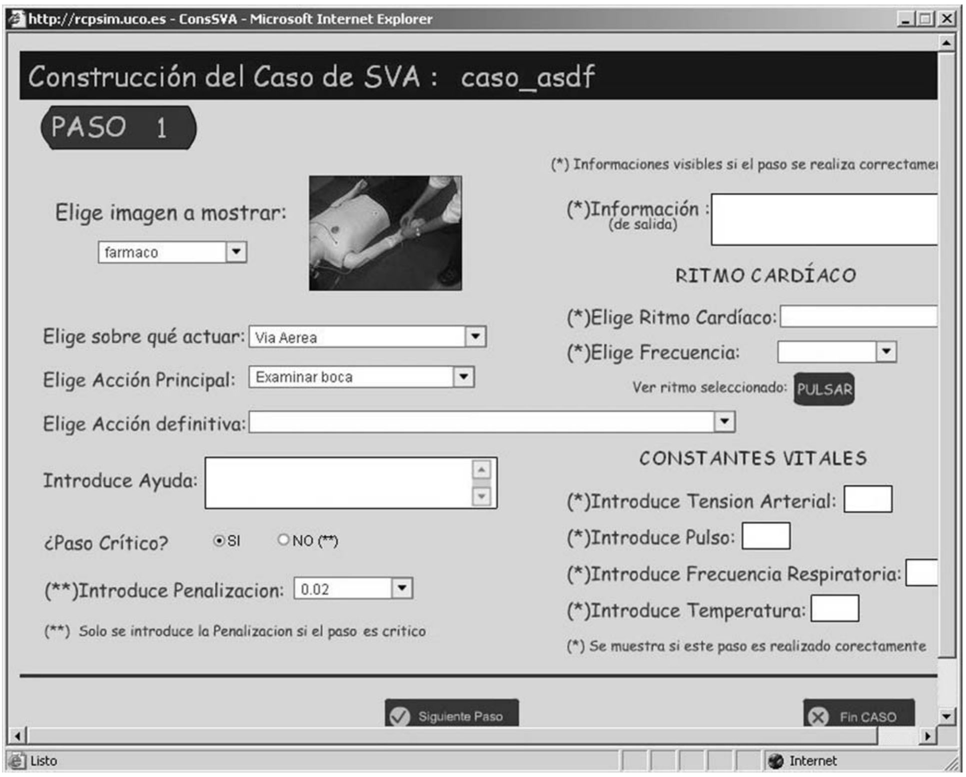
- *Creation and maintenance of exposition contents*. The teacher can create, delete, or modify the explanatory contents and assign them to courses and chapters.
- *Test development*. The teacher creates or edits multiple-choice classic and adaptive tests about life support (Section 2.5.2).
- *Case authoring*. The teacher creates or edits interactive simulations of life support cases (Fig. 2) and exam case studies (Section 2.5.3).
- *Review of students' results*. The teacher can access all students' usage information in order to see scores and times of students, error reports and detailed information about the interaction in case study execution.

Creation of new content only requires development of the exposition contents, tests and cases. In general, a single person can develop one of the courses in several weeks. However, several teachers usually share this work. The most tedious task is to build a plan for each case due to the necessity of specifying information for each step of the case.

2.2. Case simulation model

The case simulation model contains the procedural knowledge or problem solving skills about life support cases. We use a plan to represent the expert knowledge about the resolution of each case. The plan consists of initial information for the case and information about the sequence of steps with the actions an expert recommends. We differentiate between critical and moderate actions. A critical action is obligatory to perform in order to resolve the case correctly. A critical action can require previously performed moderate actions (without imposition of a specific order of execution between them). So, the teacher writes the plan in a logical execution order, but a student can resolve it using a different path, which is determined by the critical actions. Other actions can be done in any order or not at all. The teacher has to provide the following information for each case in order to build the plan (* denotes obligatory information):

- *Initial information*. Name of the case*, type of the case (example case/exam case)*, initial description of the case*, initial patient's image*, initial patient's vital signs (pulse, blood pressure, respiratory rate, temperature) and cardiac rhythm*, maximum time to execute each critical action and maximum number of allowable failed actions in each critical action.
- *Step information*. For each step: number of the step*, type of step (critical or not)*, expected correct action*, list of required actions done before, helping information about the correct


http://rcpsim.uco.es - ConsSVA - Microsoft Internet Explorer
PULSAR
Siguiente Paso
Fin CASO
Listo
Internet

Construcción del Caso de SVA : caso_asdf

PASO 1

(*) Informaciones visibles si el paso se realiza correctame

Elige imagen a mostrar: farmaco

(*)Información (de salida):

RITMO CARDÍACO

(*)Elige Ritmo Cardíaco:

(*)Elige Frecuencia:

Ver ritmo seleccionado:

Elige sobre qué actuar: Via Aerea

Elige Acción Principal: Examinar boca

Elige Acción definitiva:

Introduce Ayuda:

CONSTANTES VITALES

¿Paso Crítico? [✓ SI] NO (**)

(*)Introduce Tension Arterial:

(*)Introduce Pulso:

(**)Introduce Penalizacion: 0.02

(*)Introduce Frecuencia Respiratoria:

(*)Introduce Temperatura:

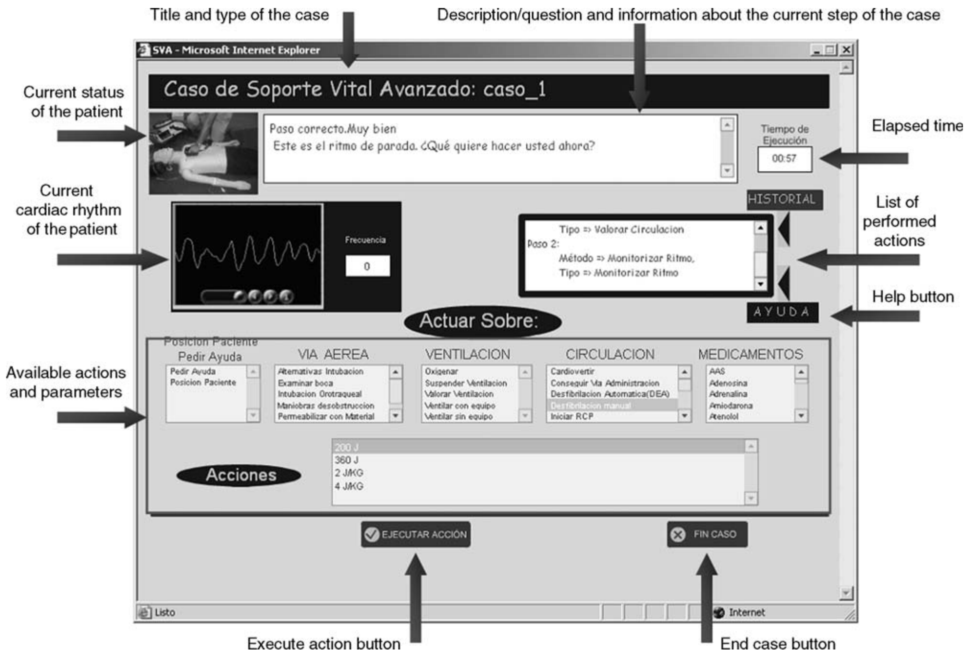
(**) Solo se introduce la Penalizacion si el paso es critico

(*) Se muestra si este paso es realizado corectamente

Figure 2 Interface of author tool to develop cases.

action, new patient's image and new patient's vital signs/cardiac rhythm if the action is done, text message to show if the executed action is correct*, and text message if not, penalization (percentage to reduce in the final score) if the action is incorrectly executed.

Students have to select the sequence of actions from a list in order to resolve the case using visual and textual information (Fig. 3). We use two basic visual components to simulate the current status of the patient in our case simulation model:


Title and type of the case
Description/question and information about the current step of the case
Current status of the patient
Elapsed time
Current cardiac rhythm of the patient
List of performed actions
Help button
Available actions and parameters
Execute action button
End case button
SVA - Microsoft Internet Explorer
HISTORIAL
AYUDA
Actuar Sobre:
Acciones
EJECUTAR ACCIÓN
FIN CASO
Listo
Internet

Caso de Soporte Vital Avanzado: caso_1

Paso correcto.Muy bien
Este es el ritmo de parada. ¿Qué quiere hacer usted ahora?

Tiempo de Ejecución 00:57

Frecuencia 0

Tipo => Valorar Circulacion
Paso 2:
Método => Monitorizar Ritmo,
Tipo => Monitorizar Ritmo

Posicion Paciente / Pedir Ayuda	VIA AEREA	VENTILACION	CIRCULACION	MEDICAMENTOS
Pedir Ayuda	Alternativas Intubacion	Oxigenar	Cardiovertir	AAS
Posicion Paciente	Examinar boca	Suspender Ventilacion	Conseguir Via Administracion	Adenosina
	Intubacion Orotraqueal	Valorar Ventilacion	Desfibrilacion Automatica(DEA)	Adrenalina
	Maniobras desobstruccion	Ventilar con equipo	Desfibrilacion manual	Amiodarona
	Permeabilizar con Material	Ventilar sin equipo	Iniciar RCP	Atenolol

200 J
360 J
2 J/KG
4 J/KG

Figure 3 Interface of case simulator in an advanced course.

Patient simulator. The patient simulator shows visual information about the physical status of the patient and the actions students have performed. We use a picture or photograph (Fig. 3 at top left) to show the current status of the patient. The teacher has to select the appropriate picture for each step when building the plan (if not, the previous image is used by default).

Monitor simulator. The monitor simulator shows the current cardiac rhythm of the patient. We use a recurrent movie (Fig. 3, below patient image) which can be paused, moved forward or rewound. Our system stores hundreds of movies with basic rhythm variants and different heart rates. The teacher has to select the appropriate rhythm and heart rate for each step to be shown to the user (if not, the previous cardiac rhythm is used by default).

2.3. Domain model

The domain model contains the declarative or conceptual knowledge of cardiac life support. We organized our domain model in a hierarchic way similar to traditional courses which consist of multiple chapters or lessons. To build our domain we used the information provided by different resuscitation guidelines [23]: adult basic life support (BLS), automated external defibrillators (AED), adult advanced life support (ALS), paediatric BLS and paediatric ALS. We organized all the information from these guidelines into 10 different chapters (Table 1). Chapters were then grouped into three courses:

Table 1 List of chapters

Number	Title
Chapter 1	Basic concepts in cardiopulmonary resuscitation
Chapter 2	Basic life support
Chapter 3	Advanced life support
Chapter 4	Advanced respiratory and circulatory assistance
Chapter 5	Pharmacological treatment and administration
Chapter 6	Arrhythmias
Chapter 7	Arrhythmias electrical treatment
Chapter 8	Paediatric basic life support
Chapter 9	Paediatric advanced life support
Chapter 10	Pre-hospital care in myocardial heart-attack

- *BLS*. Chapters 1, 2 and 8. This course includes adult BLS, and paediatric BLS guidelines, and is oriented to first responder personnel (firemen, pedestrians, etc.).
- *Intermediate life support (ILS) + AED*. Chapters 4 and 7. This course includes ILS and AED guidelines, and is oriented to emergency technicians.
- *ALS*. Chapters 1—10. This course includes all guidelines, and is oriented to nurses and physicians.

We also grouped all available actions (all the actions students can execute in the case simulation) into five categories: circulation support (53 actions), ventilation support (20 actions), airway support (33 actions), medication or drug support

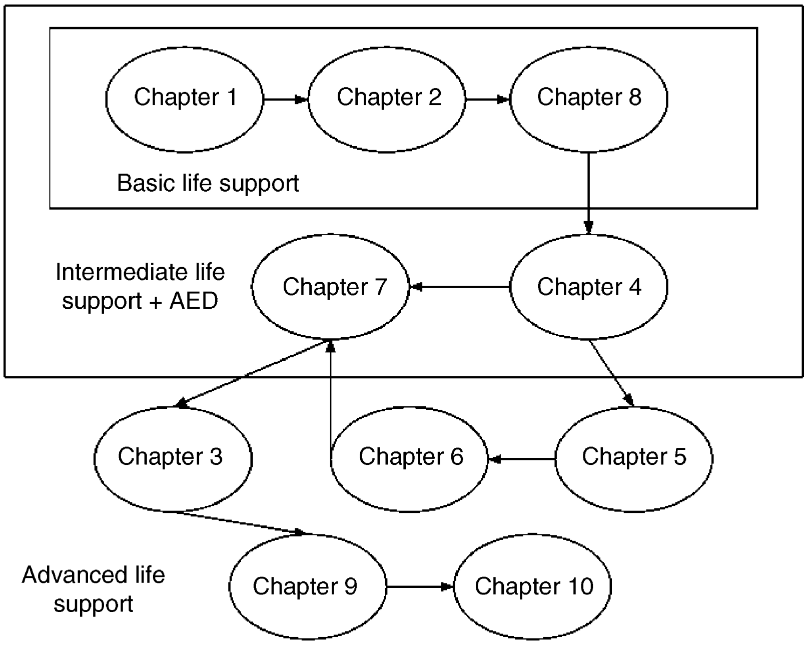


Figure 4 Relations between chapters. AED = automated external defibrillator.

(32 actions), and others (10 actions). We related actions and some elements of the case simulator interface to the three courses. The ALS case simulation interface shows all the actions and all the elements, the ILS + AED case simulation interface shows fewer actions (for example no medication is shown) and the BLS case simulation interface shows only some basic actions and the cardiac rhythm monitor is not shown.

We can also distinguish two types of relations:

- *Prerequisite relations*. Prerequisite relations between chapters (Fig. 4) are used to adapt the user navigation in the exposition contents, showing the student the appropriate chapter depending on which chapter the student visited previously.
- *Components relations*. Component relations connect educational components and the chapters (Fig. 5). Each chapter is related to one exposition content. Each test question is related to one or more chapters. Each case action is related to one chapter or several chapters. Furthermore, in the ALS course, we defined three different types of case studies depending on the level of difficulty (expert, medium and novice).

2.4. Student model

The student model contains the student's information. We did not use a complete student model but rather we use a student record with the following information:

- *Identifying information*. Name, login, password, age of student, provided by the student during the registration process.
- *Profile and execution information*. This information is automatically obtained by the system during the student's interaction and is used to adapt content delivery to each particular student and to analyze the student's learning. It includes:
  - *For each course*: name of the course, score in test, score in exam case and current learning stage.
  - *For each chapter*: name of the chapter, visited or not, time of visit, number of incorrect questions and number of incorrect actions.

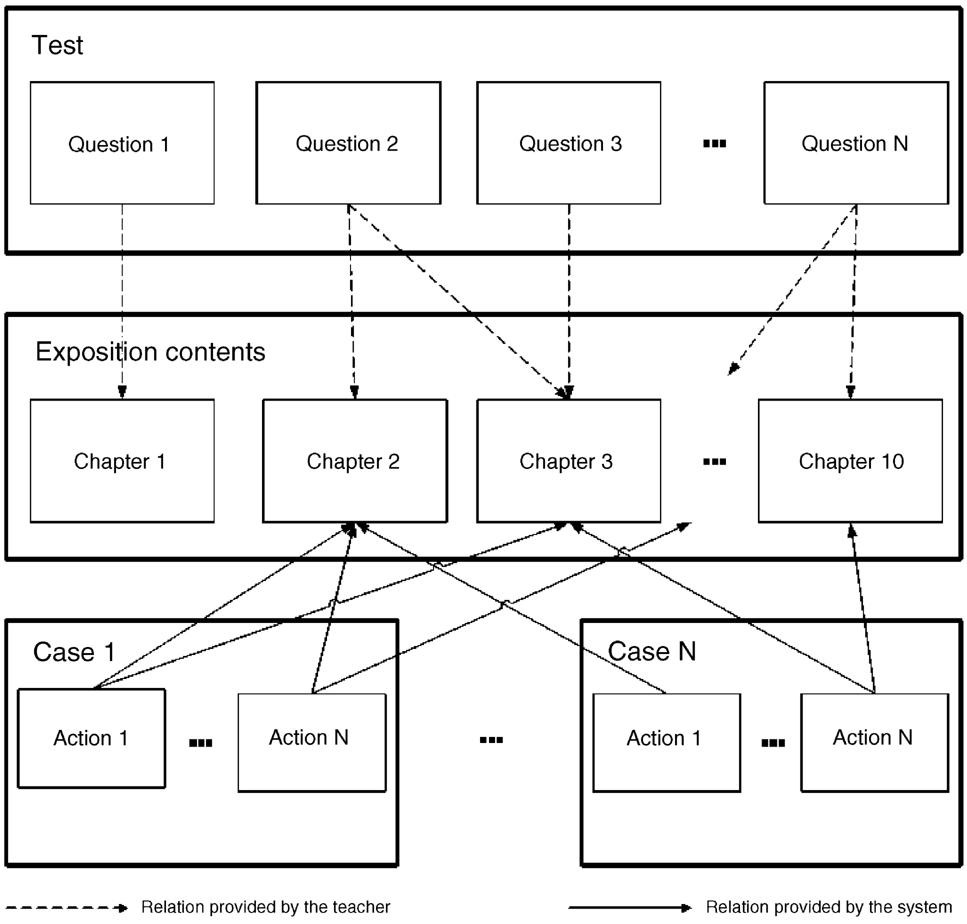


Figure 5 Relations between components.

- *For each test*: name of the test, done or not, score and time spent.
- *For each question*: name of the question, performed or not, and correctly or incorrectly answered.
- *For each case*: name of the case, done or not, score, time spent, and sequence of performed actions.
- *For each performed action*: name of the action, position number of the action in the sequence, critical or not, time spent, number of attempts, and correctly or incorrectly performed.

- *Log information*. Information about the student's session is captured by a traditional web server log. It stores information about the requested web pages, including client IP address, request date/time, page requested, HTTP code, bytes served, etc. This information is obtained by the web server during the student's interaction and can be used by the teacher to analyze the usage of the system.

2.5. Adaptation and execution model

The adaptation model or tutor model contains information about how the domain model and case simulation must be presented or adapted to each particular student. It is the engine of our system that controls the execution of the navigation in the system, the presentation of contents, the execution of tests and the execution of case simulation. The adaptation model uses a rule-based inference system (Section 3).

2.5.1. Adaptive navigation and presentation

We use a high-level adaptive navigation support technique [3] with three learning stages: studying, practicing and training (Fig. 6). The student first studies the explanatory contents and takes a computer test that evaluates his foundational knowledge. Subsequently, he practices with sample case studies until ready to evaluate his knowledge again with an exam of case studies, after which he can complete further training with more sample case studies to continue building his knowledge.

We use two guidance navigation techniques [3]. Direct guidance (hiding) sets the sequence students should do in the first reading of the chapters. The available chapters are dynamically determined based on the previously visited chapters and the relationships between chapters. Local guidance (sorting) is used after the execution of tests and cases to suggest that students review the chapters in which they made more errors.

We use an adaptive presentation technique [3] to adapt the theoretical contents of the course to the user. Alternative explanatory content pages (chapters) are chosen depending on the course the learner is registered for. Furthermore, we use adaptive presentation [3] in the case simulation interface (Fig. 3) that is different for each type of course (advanced, basic or intermediate), by hiding some interface elements and actions.

2.5.2. Test adaptation and execution

Our system can use both classic and adaptive computerized tests [20] with multiple-choice/single-answer questions. A conventional (classic) test is a sequence of simple questions and normally the same questions are shown to all examinees. The algorithm to control the execution of a classic test is very simple: it shows a sequence of questions until there are not any more questions or time expires. On the other hand a computerized adaptive test [21] is a computer-based test where the decision about presenting a question or item and finishing the test is made depending on the examinee's performance on previous answers. The general adaptive test algorithm consists of three main procedures: question selection based on the most informative item for each student; proficiency estimation of each student; and checking the finalization criteria (maximum number of questions, maximum spent time, or

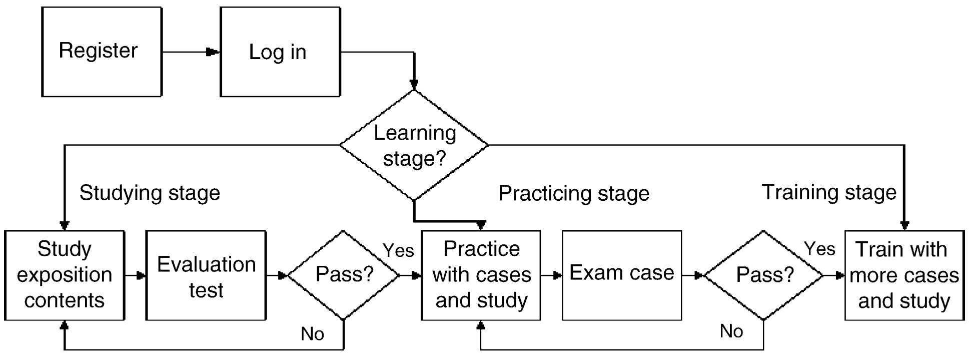


Figure 6 Step-based learning model.

if the proficiency level has passed a confidence value). Teachers build a test by setting:

- *Items or test questions*: the question or statement, the answers and the correct answer.
- *Parameters for the presentation of questions*: the order in which questions and answers are shown, whether to show or hide explanations of the answers.
- *Finalization*: the maximum time to respond and the maximum number of attempts students have to pass the test.
- *Evaluation of the test*: whether to penalize incorrect answers, whether to penalize questions without an answer.

If the test is adaptive, teachers also have to set the adaptive algorithm parameters (initial level, question selection procedure and termination criteria). We use the three-parameter item response theory model in which the probability of a correct response to a given item is a function of the examinee's proficiency and the difficulty, discrimination and guessing of the item [20]. The adaptive test has to be calibrated to estimate the item parameters [21].

2.5.3. Case execution

We have two types of case studies: sample case studies used for practice (which can be selected by the students) and exam case studies used to evaluate (which are determined by the system). Help is available only for practice cases.

The algorithm to control the execution of a case (Fig. 7) uses the plan for each case to verify if a student action is correct or not. From the case plan, the system creates a list of the current possible actions. If the user executes a correct action (one of the current possible actions and the time and number of attempts for the critical action have not been exceeded) the system presents the appropriate message and updates the status of the patient. If the user executes an action but has exceeded the time or the number of attempts, then the case ends with an incorrect resolution. If the user executes an action that is not one of the current possible actions and the time or the number of attempts has not been exceeded, then the system penalizes the student's score, increases the counter of attempts, and waits until the student executes another action. Finally, if the student executes all the actions on the list and there are no more steps in the plan, then the case ends with a correct resolution.

2.5.4. Case adaptation

An exam consists of a number of case studies. In advanced courses, cases can have three different levels of difficulty (expert, medium and novice), and the system selects the difficulty of the next case depending on whether the current case was completed correctly. Fig. 8 shows how the case adaptation selects the level of the first case depending on the score obtained by the student in the evaluation test. When the student completes the case study, the level of the next selected case depends on the score obtained in the previous case. If the obtained score is less than a lower value (specified by the teacher) then the difficulty level of the next case will be novice; if the obtained score is higher than the lower value and less than a upper value (specified by the teacher) then the difficulty level of the next case will be medium; and if the obtained score is higher than the upper value then the difficulty level of the next case will be expert. The final score is calculated as the mean value of scores (sum of all scores obtained in the cases divided by the total number of cases completed).

3. Implementation

The system was implemented using Microsoft ASP (Active Server Pages) and ASP.Net as the main programming languages, HTML (Hyper-Text Markup

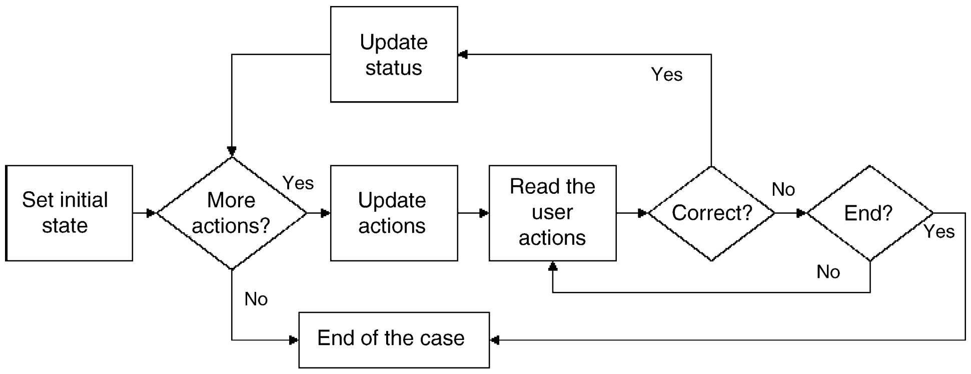


Figure 7 Algorithm to control the execution of a case.

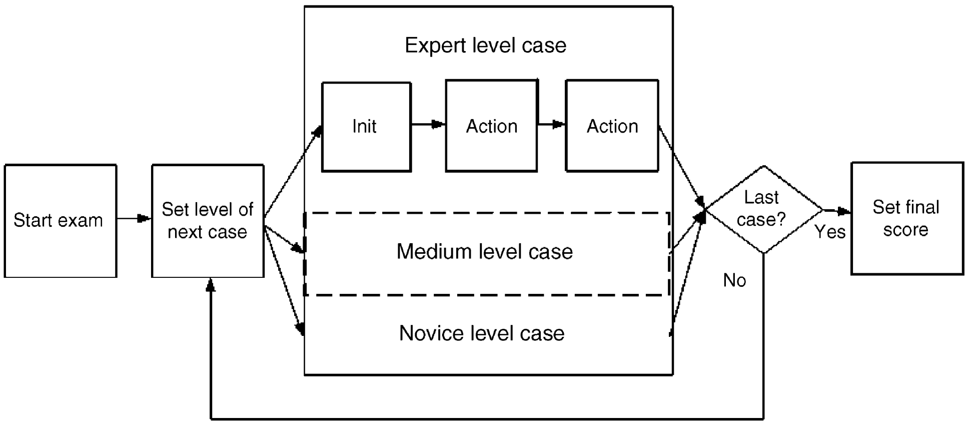


Figure 8 Case adaptation in advanced courses.

Language) to show information, Macromedia Flash in more interactive interfaces, and XML (Extensible Markup Language) to represent knowledge. We also used Microsoft Internet Information Server as web-server, MySQL as database server and Microsoft .NET Framework Developer Center.

We graphically represent the rules by means of a rhombus in the figures of the different adaptation and control algorithms. We represented the rules in XML format using RSML (Rule Structure Markup Language) [24]. RSML describes a generic rule language for encoding rules, which allows for a more flexible architecture and for the customization of procedural knowledge. These XML rules allow a general and modular implementation of the adaptive algorithms, which makes the configuration and maintenance of the system lighter. A set of rules has been included for each algorithm. The inference module is embedded in the system and is tailor-made so that the implementation fits perfectly with the system and calling other inference systems is not necessary. The inference component will load the suitable group of rules and the variables as facts. Then, it will fire those rules that satisfy the antecedents (modus ponens inference rule).

There are five different groups of rules (one for each algorithm). These rules are previously created and are automatically configured (setting some needed values, i.e. MAX TIME and MAX ACTIONS) when the author builds the educational components of the course.

4. Evaluation

We evaluated the system to determine the usefulness of the adaptation techniques by comparing the results students obtain when they used the system with the adaptive features and without them.

4.1. Study design

To determine the effect of adaptation, we implemented the non-adaptive version of the system in one course (Course N-A), and the adaptive system previously described in another one (Course A). In the non-adaptive version of the system, students could move freely between steps, see all the chapters of cardiac life support, take classic computer tests and exam case studies without any adaptation to their level. In the adaptive version of the system, students were guided by all adaptive techniques.

The parameters used to compare non-adaptive and adaptive versions were the students' results (time spent and scores obtained) and the number of items used (questions, tests, and exam cases used). These parameters were previously used to compare AH techniques with non-adaptive control hypermedia applications [25] and to compare computer-adaptive tests with conventional tests [26]. We used a between-subject comparison using the previous parameters in order to confirm the hypothesis that students using the adaptive version could obtain better or equal scores in less time.

The classic test consisted of 29 questions randomly chosen from a bank of 50 questions. Students had a time limit of 10 min to complete the test. The adaptive test used the same bank of 50 questions in which the item parameters were previously calibrated from a paper-and-pencil test carried out by more than 300 students. The adaptive test had three finalization conditions: (1) if the standard error obtained was lower than 0.33; (2) if the total number of exhibited questions was higher than 29; (3) if the student took longer than 10 min.

The classic exam case studies were made up of five cases randomly chosen from a bank of 10 cases. Students had a time limit of eight minutes to complete each case. The adaptive exam case used the same bank of cases but also the information about

Table 2 Adaptive (A) vs. non-adaptive (N-A) version

	Test		Exam	
	Version N-A	Version A	Version N-A	Version A
Number	29.0 ± 0.0	21.7 ± 2.3	5.0 ± 0.0	3.8 ± 0.3
Time	361.6 ± 61.3	274.6 ± 69.1	1382.7 ± 56.8	1096.0 ± 73.0
Score	60.2 ± 7.8	62.6 ± 7.0	51.3 ± 7.6	53.2 ± 7.6

the case's level of difficulty. The adaptive exam was finished in any of the following three circumstances: (1) if the student successfully completed two cases of the same difficulty level; (2) if more than five cases were completed; (3) if the student took longer than 40 min.

4.2. Setting

The system was installed on an Intranet server to be used as a complementary tool within two identical ALS training courses at the Jaen Hospital in Spain, separated by several weeks. They were traditional adult ALS presence-taught courses lasting 3 days and aimed at physicians working in hospital emergency services. A new e-learning section, in which students use our web-based adaptive training simulator system, was added to the courses. Over the 3 days, several professional experts in emergency medicine led different workshops about adult ALS. Afterwards, all students used the web-based system in order to continue learning and training in life support case resolution. The number of people in each course was 29 with a mean age of 34 years.

4.3. Results

In Table 2, we show the mean value and the confidence interval (95%) of the time taken (in seconds) to complete tests/exams, the number of questions/cases attempted and the final score (in percentage) in tests/exam. Two important differences were observed when comparing both versions. First, as we can see in the first table row, there was a reduction in the total number of questions/cases used in the adaptive version versus the non-adaptive version (a single-factor ANOVA revealed significant differences in both cases, $F = 37.8$ in tests and $F = 54.4$ in exams, $P < 0.05$ in both cases). In the non-adaptive version each student must take exactly 29 questions and five cases. Second, we can see a reduction of the time needed to complete the tests/exams in the adaptive version versus the non-adaptive version (single-factor ANOVA $F = 3.4$ in tests, and $F = 36.9$ in exams) precisely due to the reduction of questions/cases. Despite the decrease in time and questions, the final score obtained in tests/exams in both versions is equivalent (a single-factor ANOVA yields an F value of 0.66 in test scores and an F value of 0.13 in exam scores).

5. Conclusions and future work

We obtained an improvement of the students' productivity using the adaptive version of the system and similar students' results to the non-adaptive version. The results suggest that adaptation can mitigate some of the time-consuming and inefficient aspects of non-adaptive systems.

We are now working on the application of different data mining techniques to provide intelligent feedback (not simple reports of frequent errors) to the authors of the system [27]. Data mining is a multidisciplinary area, which attempts to discover new interesting and useful knowledge. Knowledge discovery methods allow us to discover new knowledge by providing feedback to authors based on students' usage data. We are researching data mining tasks (association, classification, clustering, etc.) that could be applied to improve our system. Our objective is to assist teachers in detecting possible errors or problems in student learning, and shortcomings and improvements to the system, for instance, to classify students in groups based on their scores, to discover new adaptation rules of the system from students' usage data, or to verify if the difficulty of the cases is well established.

Acknowledgement

The authors gratefully acknowledge the financial support provided by the Spanish Department of Research of the Ministry of Science and Technology under TIC2002-04036-C05-02 and TIN2005-08386-C05-02 Projects.

References

[1] Lillehaug S, Lajoie S. AI in medical education—another grand challenge for medical informatics. Artif Intell Med 1998; 12:197—225.

[2] Haag M, Maylein L, Leven FJ, Tonshoff B, Haux R. Web-based training: a new paradigm in computer-assisted instruction in medicine. Int J Med Inf 1999;53(1):79—90.

[3] Brusilovsky P. Adaptive educational hypermedia. In: Ruokamo, et al., editors. Proceedings of 10th international PEG conference. 2001. p. 8—12.

[4] Sleeman D, Brown JS. Intelligent tutoring systems New York: Academic Press; 1997.

[5] de Bra P, Aroyo L, Chepegin V. The next big thing: adaptive web-based systems. J Digital Inform 2001;5:1.

[6] Eliot C, Neiman D, Lamar M. Medtec: a web-based intelligent tutor for basic anatomy. In: Lobodzinski S, Tomek I, editors. Proceedings of the WebNet'97 world conference of the WWW. 1997. p. 1—5.

[7] Clancey W. Overview of Guidon. J Computer-based Instruct 1983;10:8—14.

[8] Beaumont I. User modeling in the interactive anatomy tutoring system Anatom-Tutor. User Model User-Adapted Inter J Personal Res 1994;4(1):21—45.

[9] Suebnukarn S, Haddawy P. A collaborative intelligent tutoring system for medical problem-based learning. In: Nunes N, Rich C, editors. International conference on intelligent user interfaces. 2004. p. 140—21.

[10] Crowley RS, Medvedeva O, Jukic D. SlideTutor—a model-tracing intelligent tutoring system for teaching microscopic diagnosis. In: Hoppe U, Verdejo R, Kay F, editors. Proceedings of the 11th international conference on artificial intelligence in education. 2003. p. 157—64.

[11] Rovick A, Michael J. CIRCSIM: IBM-PC computer teaching exercise on blood pressure regulation. In: MacIntosh FC, editor. Proceedings of the XXX congress of international union of physiological sciences. 1986. p. 138—9.

[12] Ramakrishnan S, Hmelo CE, Day R, Shirey W, Huang Q. Integration of novice user interface into a professional modelling tool. In: Hanley, Belfus, editors. American medical informatics association annual symposium. 1998. p. 6780—682.

[13] Kinshuk RO, Rashev R, Simm H. Interactive simulation based tutoring system with intelligent assistance for medical education. In: Ottmann T, Tomek I, editors. Proceedings of the ED-MEDIA. 1998. p. 715—20.

[14] Eliot C, Williams K, Woolf B. An intelligent learning environment for advanced cardiac life support. In: Cimino JJ, editor. Proceedings of the AMIA annual fall symposium. 1996. p. 7—11.

[15] Stansfield S, Shawver D, Sobel A, Prasad M, Tapia L. Design and implementation of a virtual reality system and its application to training medical first responders. Presence-Teleoperators Virtual Environ 2000;9(6):524—56.

[16] Viciana-Abad R, Reyes-Lecuona A. Patient modelling using expert systems for medical training simulations based on virtual reality. In: Goebel M, Watson B, editors. International conference on virtual reality. 2005. p. 10—9.

[17] Riley R. In: Keneally J, editor. The use of simulators in medicine. Melbourne, Australia: The Australian and New Zealand College of Anaesthetists (ANZCA); 2000. p. 1—12.

[18] http://www.cprsim.com (accessed: 23 January 2006).

[19] http://mdchoice.com/pt/simulator/simulatorlink.asp (accessed: 23 January 2006).

[20] Romero C, Ventura S, De Bra P. An authoring tool for web-based adaptive and classic tests. In: Nall J, Robson R, editors. E-learn: world conference on e-learning in corporate, government, healthcare and higher education. 2004. p. 1740—177.

[21] Wainer H. Computerized adaptive testing: a premier New Jersey: Lawrence Erlbaum Associates; 2000.

[22] Rios A, Millan E, Trella M, Conejo R. Internet based evaluation system. In: Lajoie S, Vivet M, editors. Artificial intelligence in education: open learning environments. Amsterdam: IOS Press; 1999. p. 387—94.

[23] Monsieurs K, Handley A, Bossaert L, Latorre F, Nolan J, Robertson C, et al. The resuscitation guidelines. Resuscitation 2001;48:199—324.

[24] Lee JK, Sohn MM. Extensible rule markup language—toward the intelligent web platform. Commun ACM 2003;46:9—64.

[25] Hothi J, Hall W. An evaluation of adapted hypermedia techniques using static user modelling. In: Brusilovsky P, de Bra P, editors. Proceedings of the second workshop on adaptive hypertext and hypermedia. 1998. p. 45—50.

[26] Madsen H. Evaluating a computer-adaptive ESL placement test. CALICO 1998;4(2):41—50.

[27] Romero C, Ventura S, De Bra P. Knowledge discovery with genetic programming for providing feedback to courseware author. User Model User-Adapt Inter J Personal Res 2004; 14(5):425—64.