

Integration of an Intelligent Tutoring System in a Magnetic Resonance Simulator for Education: Technical Feasibility and User Experience

Daniel Treceño-Fernández^a, Juan Calabia-del-Campo^b, Miguel L. Bote-Lorenzo^a,
Eduardo Gómez Sánchez^a, Rodrigo de Luis García^a, Carlos Alberola-López^{a,*}

^a*ETSI Telecomunicación, Universidad de Valladolid, 47011 Valladolid, Spain (e-mail: dtrefer@lpi.tel.uva.es, {migbot,edugom,rodlui,carlos}@tel.uva.es).*

^b*Juan Calabia-del-Campo, Hospital Clínico Universitario, Valladolid, Spain (e-mail:juancalabia@gmail.com).*

Abstract

Background and Objective: In this paper we propose to include an intelligent tutoring system (ITS) within a magnetic resonance (MR) simulator that has been developed in house. With this, we intend to measure the impact, in terms of user experience, of including an ITS in our simulator.

Methods: We thoroughly describe the integration procedure and we have tested the benefits of this integration by means of two actual educational experiences, with one of them using the simulator as a standalone tool, and the other with the joint use of simulator+ITS. The experiences have consisted of two online courses with a number of students around 180 in both of them, where measurements of usability, perceived utility and likelihood to recommend were collected.

Results: We have observed that the three measurements improved noticeably in the second course with respect to the first one. In addition, quantitative measurements are complemented with comments in free text format directly provided by the students. Results have provided evidence on the benefits of integrating an ITS in terms of quantitative user experience, as well as qualitative comparative comments directly by students of both courses.

Conclusions:

*Corresponding author.

This is the first time that an ITS is used within the scope of MR simulation for training purposes. Benefits of integrating an ITS within an MR simulator have been evaluated in terms of user experience, with satisfactory comparative results.

Keywords: Magnetic Resonance Imaging MRI, Simulator System, Medical training/educational tool, ITS, Radiographers training

1. Introduction

Magnetic Resonance (MR) Imaging (MRI) is one of the most powerful medical imaging modalities available today; its rich contrast in soft tissue as well as its tremendous versatility make it the preferred technique in many clinical situations. However, operating a MRI scanner is not an easy task and several factors contribute to these difficulties. First, the fundamentals of MRI lie on advanced concepts of nuclear physics, electromagnetism and signal processing; second, MRI acquisition heavily depends on a large number of parameters, which should be fine-tuned for optimal quality and, specifically, to avoid the onset of artifacts. Furthermore, clinical demands and high costs imply extensive scanner use, so hands-on sessions for practitioners can hardly be afforded.

Being this the case, the development of computerized simulators for training in MRI seems a solid educational alternative. If properly designed and implemented, such simulators would provide a deeper understanding of the elements involved, as well as extensive practicing opportunities for trainees prior, or in addition, to the interaction with a real scanner. Accordingly, a number of MRI simulators have been proposed in the literature, some examples of which are [1–13]. Leaving aside a discussion on which of the simulators mentioned above is more appropriate as a general purpose educational tool in MRI, a key question in this field that remains formally unanswered is how to quantify the actual usefulness of such a simulator in an actual classroom experience, i.e., whether the inclusion of a simulator in the learning methodology improves educational results. We have recently contributed to this field by designing and developing an in-house simulation tool [14] and quantifying this improvement; our evidence provides a positive answer to this question [15].

However, the observation of the educational experiences with our simulator indicates that learning MRI is intrinsically complex, so any additional guidance that the students could be provided with would be very much valuable. This additional guidance can be offered in a number of ways, such as audiovisual material, exercises to be executed and the like. Actually, the incorporation of an intelligent tutoring system (ITS) in an MR educational simulator would imply a major breakthrough for trainees learning support.

In the education community, usage of technology in tutoring activities has a long record. The system called SCHOLAR [16] traces back to 1970, and it was employed in South Africa to help students learn Geography. The term *Intelligent Tutoring System* (ITS) was coined in 1982 [17], and it was defined as the application of artificial intelligence techniques to the adaptive execution of tutorial and learning activities. Since then, the field has been quite active, probably spurred by many factors such as their success in educational experiences and the ever growing capabilities of both hardware and software.

In this paper our contribution is twofold, namely, both technological and educational; as for the former, we have satisfactorily included an ITS within our simulator, and the details of this inclusion will be provided. The interested reader will find in the paper sufficient information to carry out a similar adaptation of a well-known ITS toolkit to their specific application domain with no alteration of the code of any of them. To the best of our knowledge, this is the first MR simulator that incorporates an ITS. As for the latter, albeit preliminarily, we have quantified the impact of this inclusion by means of several well-known measures of user experience directly calculated from the information provided by the students of an on-line course recently given for Spanish-speaking radiographers; this course has been certified by an external organism (see Section 3.3.1 for details). The figures obtained indicate that the ITS is indeed an amenable tool that increases students satisfaction about the simulator, as quantified by the three parameters we have measured, namely, usability, perceived utility and likelihood to recommend.

The paper is structured as follows: Section 2 provides an overview of related work in terms of ITSs used in different educational application domains. Section 3 describes

materials and methods. Specifically, section 3.1 provides an overview of the MR simulator. Then, we describe how the integration of the ITS with our simulator is carried out. This is done in section 3.2; this section starts with a justification of the need of tools that facilitate the process of ITS design and development and we further explore the tool we have chosen for our integration; the motivation for our selection is also included (subsection 3.2.1); this is followed by respective subsections on analysis, design and implementation. The section concludes with two examples of use to illustrate the ITS from a perspective of both the expert and the student (section 3.2.5). Section 3.3 is dedicated to the evaluation of the student experience; this section is divided in two subsections, respectively dedicated to describe the two educational experiences in which the simulator has been used and the measurement instrument. Results are described in Section 4, which are then discussed in Section 5. Concluding remarks as well as future work are summarized in Section 6. Finally, the paper includes a section with supplementary material, in which we provide some pieces of code that can help the interested reader understand in detail some software solutions.

2. Related Work

As stated in the previous section, the onset of ITS took place in 1970 while the name itself was established twelve years later. Since then, ITSs have been proposed for very different application domains, such as military environments, with *BILAT* [18], *Helicopter PilotUS* [19] or *Tactical Action Officer* [20]. Many other civilian domains have been explored, such as helping children to learn how to read, with the *Project Listen* [21], mathematical applications, with the *Cognitive Tutor Algebra I* [22] or *Wayang Outpost* [23], applications in physics with *Andes* [24] or even an application in microbiology with *Crystal Island* [25] that makes use of three dimensional environments and game-based learning applied to that discipline.

As of today, a recent review [26] summarizes the landscape in this field in the interval between 2007 and 2017; 53 papers are surveyed. Most of them ($\approx 38\%$) focus on computer applications while there is a tie between health/medical and math-related topics ($\approx 15\%$). ITSs are indeed shown to be effective, with results very much

comparable with human tutors [27] or even higher [28] than groupwise human-based instruction, non ITS computer-based education and text and exercise books.

Some examples close to the topic we deal with in this paper are the following: a simulator of virtual anesthesia is described in [29]. It is clearly oriented to medical personnel. In a later publication [30], the authors evaluate the decisions made by the system, how information is handled and how tutor adaptation to each student profile is carried out. Another medical ITS is described in [31]; in this case, the system intends to provide advice to develop skills for clinical assessment and decision making. The authors evaluated student skills after a training period where students received advice either from human experts or from an ITS. Results showed very similar grades (4.2 and 4.4 respectively in a 5-point scale) with one or the other modality. As for learning of computer applications, [33] describes a massive experiment (1500 students) with an on-line platform to teach how to use Microsoft Excel and PowerPoint. Results reported showed a 26% grade increase with respect to those students that did not use the adaptive ITS.

However, despite the concern of the educational community in educational platforms where images play a key role [32], the field of educational MR simulation seems unexplored as far as ITS technology is concerned.

3. Materials and Methods

3.1. The MRI Simulator

We have designed and implemented an MRI simulator intended for educational purposes, where the main targeted audience is the community of MR radiographers (technologists) although the tool has been satisfactorily employed in two on-line courses with different student backgrounds (see details in section 3.3.1). In this section we provide a brief overview on the simulator, aimed at making the paper self-contained. The reader is referred to [14] for further details.

3.1.1. Main features

The system is grounded on two non-functional requirements, namely, (a) short simulation times are mandatory so that action/reaction is possible in acceptable times for an

educational session; (b) the simulator should be of easy access/installation and should be able to work on a wide range of platforms.

The simulator, on the other side, resembles an actual (albeit vendor-independent) MR console and the operator is allowed to carry out the customary actions made by radiographers in their daily practice. These include:

- Entering patient information, patient positioning and coil selection.
- Creating protocols as a series of customary sequences.
- Basic acquisition parameter tuning, such as TE (echo time), TR (repetition time) and, where applicable, IR (inversion time), flip angle, ETL (echo train length) and others.
- Geometrical planning, which includes slice orientation, FOV (field of view) determination, slice thickness, slice separation and selection of phase/frequency encoding directions.

In the simulator it is also possible to activate/deactivate the presence of image artifacts as well as to define different user profiles (administrator/teacher/student). Hence, the tool allows the users both to carry out tasks customarily executed by MR technologists but also to practice on the influence of MR parameters in the final image quality.

3.1.2. Architecture and implementation

Based on the foregoing requirements, we have opted for a simple simulation model, consisting in evaluating mathematical expressions of well-known sequences which are then corrupted with artifacts; k-space access is needed to this end. Rigorous solutions of Bloch equations [34] have been avoided for interactivity issues. As for ease of access and installation, we have opted for a Web-based application.

The system has been designed with a client-server architecture; the server, follows a service-oriented architecture (SOA), where services are depicted in figure 1. The simulation service (Figure 1) uses the simple object access protocol (SOAP) given the need to exchange requests and responses with the client. The rest of the services expose a

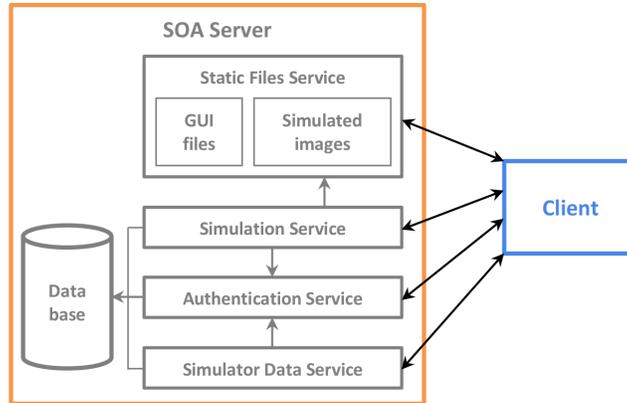


Figure 1: Generic architecture of the simulator consisting of several services that, in our deployment, run in one SOA server.

representational state transfer (REST) application program interface (API). The server has been programmed in Python using the Django framework [35]. Simulations are performed using C++ and the ITK library; interaction with Python is achieved through a wrapper.

On the client side, the interface has a component-oriented design, where each interface element consists of one or several components that follow the Model - View - ViewModel (MVVM) pattern [36]. The graphical user interface (GUI) has been implemented with AngularJS.

3.1.3. Graphical User Interface

The main interface is shown in Figure 2, where the user needs to: enter patient information (Figure 2-b); select patient position and coil (Figure 2-d); when this action is taken, the icon in 2-c will vary accordingly; select a protocol from a menu that pops when the protocol button is clicked (Figure 2-e). Once selected, the protocol is loaded in Figure 2-i; this panel contains the information of each of the pulse sequences that make up a certain protocol.

As a rule, for each sequence the user should select the relevant parameters through panels e, j, k, l, m in Figure 2, and then carry out the appropriate geometrical planning (see Figure 5c). Finally, the scan button should be pressed (Figure 2-n). At this moment



Figure 2: Simulator main interface. Superimposed letters are used to guide the explanation in the main text.

the data of the interface (i.e. the client) is sent to the server, which will carry out the simulation and return a volume for its visualization in panel in Figure 2-g. A more detailed visualization can be obtained by pressing the button in Figure 2-h; this action opens another panel where the image is represented with higher resolution.

Some other panels offer additional functionality; this is the case of 2-e, where imaging options such as a *no phase wrap* acquisition or a *shimming* procedure can be selected.

Finally, following the requirements enumerated in section 3.1.1, we have included a panel to activate/deactivate different image artifacts, to select different hardware properties (field strength or field inhomogeneity, for instance), and to choose a specific case to be simulated (different anatomical regions and/or different pathologies). Most of these utilities are accessible using the buttons in the panel in Figure 2-a.

3.2. Integration of an ITS in the MRI simulator

3.2.1. CTAT

As indicated in the introduction, ITSs have a long tradition and have been used in different educational experiences with satisfactory results. Different experiments have quantified their relative efficiency with respect to other more traditional educational methodologies; as a consequence of their success, a large number of papers have been

published in this area in the last ten years. Nevertheless, as referred to above, we are not aware that an ITS has ever been integrated in an MR educational tool.

The design and implementation of an ITS, however, are not easy tasks, since they require deep knowledge of two application domains, namely, psychopedagogy and computer science. Actually, estimates between 200 [37] and 300 hours [38] of technical effort per hour of instruction have been reported. Consequently, a number of *authoring tools* (AT) have been developed to alleviate this intricate tutoring building process. Some of these tools are ASPIRE [39], CTAT [40], xPST [41], ASTUS [42], AutoTutor [43], SitPed [44] or GIFT [45], to mention a few.

In this paper we have selected CTAT; this ITS consists of a set of tools that allow the designer to carry out an ITS that stems from a rule-based cognitive model, i.e., an ITS that is grounded on a number of rules that students should follow in order to carry out a predefined task [46]. More specifically, CTAT provides an easy way to create a *model tracing* ITS, i.e., the model traces the interaction of the learner with the system and, by evaluating the defined rules, the ITS follows the learners progression and is thus capable of suggesting the student the next step to follow, to show a demo or to provide further feedback. As can be inferred, such an ITS would fit in well-defined problems, where the solution space is somewhat limited. This is the case for MR acquisition, where the radiographer faces a number of alternatives in order to maximize image quality and to minimize acquisition time. Consequently, the ITS typology that CTAT provides suits our requirements. In addition, this toolkit permits a low coupling degree between our interface and the ITS itself. Since we have decided to incorporate an ITS to our [already existing] simulation tool, we have to select an AT that allows us to easily connect its components with those of the simulator; moreover, this toolkit is well documented and provides support for execution in a Web browser.

3.2.2. Analysis

As previously stated, our objective is to integrate CTAT within our simulation tool; the integration should take place on the client side. This integration will be designed to comply with a number of functional and non-functional requirements. As for the latter:

1. The ITS should not affect regular simulator performance and should hardly affect

its efficiency.

2. Development of the ITS and the simulator should be independent of each other.

These two requirements seem quite obvious since the ITS is intended to help the students, so the simulator interactivity should be maintained to comply with its own non-functional requirements (as stated in section 3.1.1); otherwise, the ITS itself may prevent students from using the simulator, a consequence that would make this effort worthless. In addition, the simulator is an ongoing project so additional functionality may be incorporated in the simulator at any time; hence, the ITS should not be an obstacle for simulator improvement.

As for the functional requirements:

1. Components to be registered by the ITS should be selectable.
2. The ITS should only be active when it is specifically demanded.
3. Topics and specific exercises within the topics should be selectable; the problem statement should be available to the student at any time.

This first requirement allows the expert to create the exercise; for a particular exercise not all the simulator parameters are of interest, but those targeted by the expert, which should be selected to provide appropriate feedback during the exercise execution. In addition, the ITS should only be available when it is specifically invoked, i.e., the simulator may or may not be used with the ITS at expert will, depending of the exercise to be carried out or, complementarily, to the skills of the targeted audience. Compliance with the third requirement allows instructors to organize their learning activities for better overall comprehension of the course. Needless to say, the appropriate reading material should be at disposal of the student for exercise completion.

3.2.3. *Design*

Figure 3 provides an overview of the solution we propose; this figure is an adapted version of Figure 6 in [47]; in our case, the simulator plays the role of the student

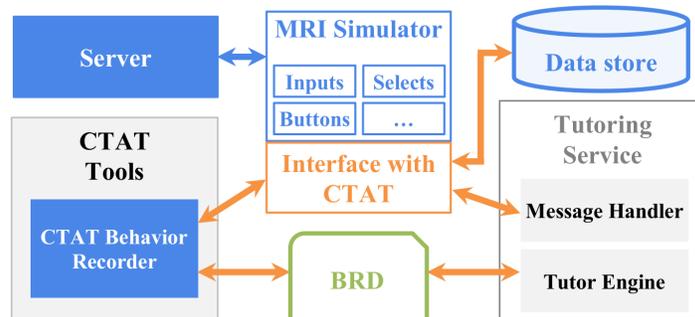


Figure 3: Schematic solution proposed to integrate CTAT with an existing simulation tool.

simulator in [47] and the key of the method is the intermediate layer (labelled as *Interface with CTAT*) that should be able to handle the communication between the CTAT Behavior Recorder, the tutoring services and our simulator.

The CTAT Behavior Recorder is a tool that provides support to the expert to create an exercise, where each step expected from the student is defined and a state diagram, together with associated transitions, is created to mimic the step sequence that the student should follow. This sequence is accompanied by a number of messages (referred to as feedback or hints). The sequences then translates to the Behavior Recorder Graph (BRD), which is an XML file that fully defines the problem, the solution and the messages to be provided. CTAT also incorporates a module that carries out the tutoring activities themselves (referred to as *Tutoring Service* in the figure); this module, to put it short, monitors the interface used by the student and decides the actions to take after each student step. The actions may vary from changing the colors of some elements in the interface as a result of the student interaction (say, green for correct, red for incorrect or yellow to highlight some areas of the interface) or showing messages in the informative panel. These actions are carried out by the tutor engine out of the information in the BRD and the steps that the student takes.

CTAT runs in a two-step procedure; the first stage is an initialization process where all the components that CTAT will use are created. This early component creation poses some difficulties for registering dynamic components; we will come back to this issue later on. Next stage has to do with the problem itself either as in *Authoring* mode

or as in *Resolution* mode. In the former, the problem is designed by the expert and we may in turn split it in three phases, namely, (a) definition of initial state, (b) BRD creation and (c) model testing. As for the latter, i.e., the mode used by students, the components send and receive information concerning the ITS.

The solution proposed in Figure 3 allows the simulator and the ITS to run in parallel and interactions are limited to information exchange. Consequently, the two non-functional requirements are *per-se* satisfied as long as the information overhead does not give rise to an unmanageable burden. Specifically, the interface module between our simulator and CTAT is created by means of the two following actions:

1. To create a procedure to share variables among the CTAT components as well as to provide tools that the programmer may use to configure, initialize and track CTAT.
2. To create a procedure to override the functionality of some CTAT components so that they behave as we expect in our simulator. This override may be done dynamically despite CTAT components are loaded at an early execution stage.

Consequently, the interface elements have to know the state of the ITS as well as additional related details; they are summarized in a number of parameters that should be available to all. How this is actually carried out depends on the functionality provided by the framework that the programmer may use. The next section provides additional information on our implementation.

Overall, complying with the two functional requirements related to selectable components is a matter of simulator interface as well as of passing this information to CTAT. As for either to activate or not the ITS, we have set two URLs, one for the ITS+simulator and the other for the simulator in isolation.

3.2.4. *Implementation Details*

With respect to the CTAT early component creation, it is the programmer task to guarantee that CTAT components carry out the appropriate functionality related to the simulator, i.e., to achieve that the effect of the ITS is that expected by the simulator. Therefore, two actions are mandatory: (a) the component load ordering should be such

that guarantees the correct CTAT functionality; in our case, since we have used AngularJS as the framework on which the simulator is built, these components should be loaded prior to the CTAT loading; (b) before CTAT is loaded, the overall system has to know whether CTAT is loaded in *Authoring* mode or in *Resolution* mode. For the former, each of the three possible phases described in Section 3.2.3 should be taken in consideration.

If CTAT starts in Authoring mode, the CTAT Behavior Recorder must be initialized, then the simulator is launched and communication between both should exist. If this is the case, the initial state must be defined. To this end, the main components of the simulator, together with the presence or absence of artifacts, are to be set. Then, the components to be registered in CTAT must be selected; when CTAT is invoked, those components should be visible. Next step is BRD creation; this translates itself into a normal simulation exercise, the steps of which are simultaneously registered by the CTAT Behavior Recorder although limited to the selected components. These components are created by means of AngularJS directives.

Dynamic components are frequent in our simulator and this collides with the policy of early creation on which CTAT is grounded. Specifically, CTAT assumes that all its components are defined before the function `initTutor()` is invoked. CTAT, to this end, proposes an *ad-hoc* solution for dynamic components which, put it short, consists in overriding the function to make it global and to invoke it once all the components to be registered have been generated. In our case, using AngularJS directives, components are handled naturally, regardless of their static or dynamic character. Additional explanations on specific code are provided as supplementary material; although we describe an AngularJS implementation, the reader may adapt these particular examples to their specificities.

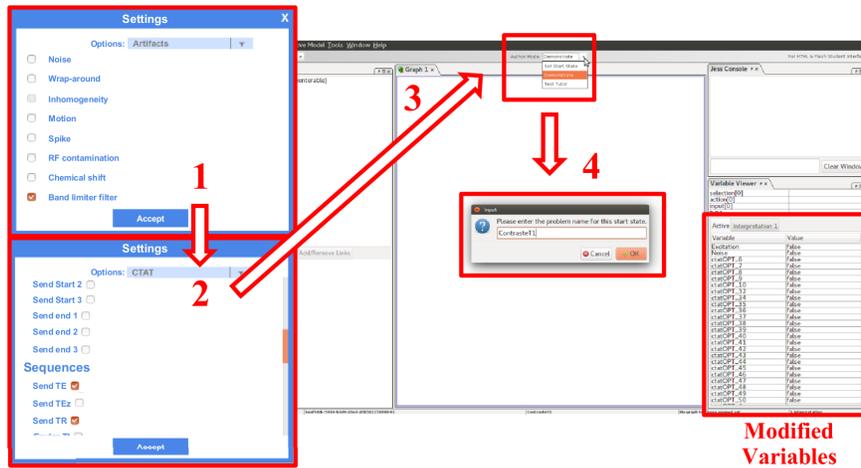
As for the problem statement, we will use one of the panels shown in Figure 2; specifically, panel a), which is where the main administrative functionality is located, seems the right place. Needless to say, the panel should sense whether the ITS is active to show/hide the problem statement accordingly.

3.2.5. Examples of use

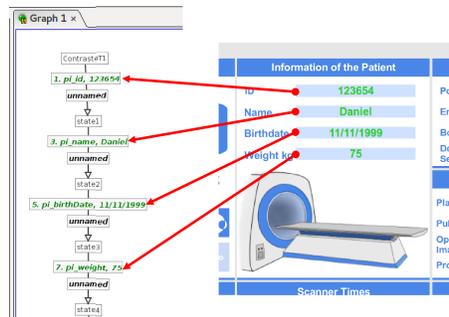
Authoring mode. Figure 4 shows an example of how a problem is created in our simulator. First, the initial state should be set (Figure 4a); to this end, we should determine the artifacts that will contaminate our acquisition (Figure 4a-1) as well as the components to be registered (Figure 4a-2). Assume that this example is related to getting some sort of contrast in MR. Accordingly, the parameters to be registered should be TE and TR (see the tick marks in the figure). BRD creation is the next step to take (see Figure 4a-3). To this end, we switch to *Demonstrate* and then we are prompted to set a name to the problem with the initial state just defined (Figure 4a-4); for this example, we will use the name *TI Contrast*. The variables that have been modified in the initial state they now show up as registered variables on the right hand side of the figure. The BRD is created by sequentially executing steps in the simulator (Figure 4b); as these steps are taken, the CTAT Behavior Recorder generates the diagram; in this case, the steps consists in entering patient personal data.

Resolution mode. Figure 5 shows the GUI associated to the simulator when the ITS is active; as can be seen in the banner on the left hand side of figure 5a (pointed by the red arrow), an option to select is labelled as *Exp. Tutor*; if this option is selected, the panel shown in figure 5b is shown by the application; this panel is used by the student to select both the course of interest amongst those available and the precise exercise in that course. After pressing the *Begin* button, a new panel is shown, so both the problem statement as well as additional graphical support can be provided (see Figure 5c). This figure also shows the ITS panel (labelled as *Tutor dialog box*); this is used by the ITS to provide messages to the student as the problem is executed.

Figure 6 shows an example of the ITS feedback. Figure 6a shows an incorrect entry to the particular field (in this case, *Id*) as well as the correct answer. The former is written in red while the latter uses green color. Other feedback (not shown in the figure) may be provided in yellow, indicating the student the next part of the GUI to work on. Figure 6b shows text feedback provided by the ITS on the Tutor dialog box. For the example, this the result of clicking on the button labelled as *Hint*. Complementarily, if a mistake is committed by the student, the ITS will prompt messages on that panel for



(a) Definition of initial state



(b) Graphical definition of student expected behavior

Figure 4: Creation of a problem with CTAT.

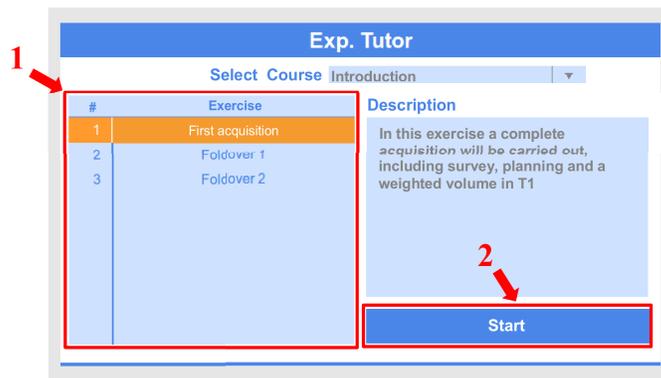
guidance.

3.3. Evaluation of user experience

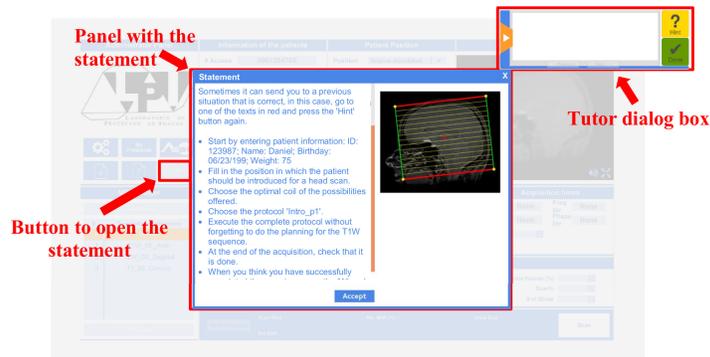
In this section we provide the details on how we have measured the impact, in terms of *user experience*, of including an ITS in our simulator. To this end, we have gathered together the results we have collected in two actual educational experiences, carried out in 2018 and 2019 respectively. In the former, the simulator was used as a standalone tool; in the latter, the simulator was complemented with the ITS. Comparative results will provide us with conclusions on the inclusion of the ITS in our simulator as for



(a) GUI as seen by the student with ITS active



(b) Panel selector of exercise to be tutored.



(c) Panels of problem statement (center) and ITS message (upper rightmost corner).

Figure 5: Creation of a problem with CTAT.

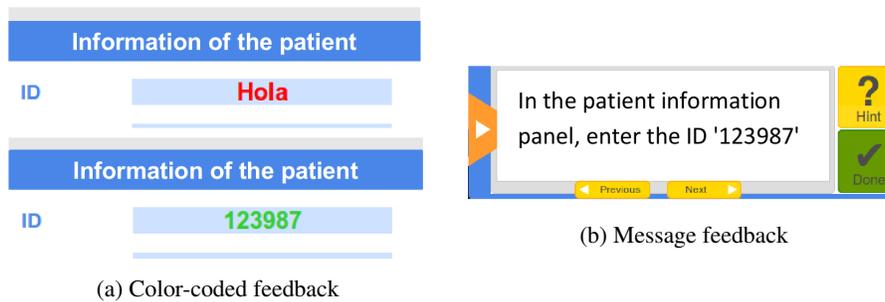


Figure 6: Examples of CTAT feedback

usability, perceived utility and likelihood to recommend.

3.3.1. Educational experiences

As previously stated, the simulator has been used in two experiences. Details follow:

1. On-line course in 2018: an eight week on-line course starting April 30th, 2018, with estimated dedication of 5-6 hours per week. The number of students was 179, with 72.1% radiologists, 21.2% radiographers (the rest were engineers and other profiles). Students were asked about general computer skills and 51% answered “basic”, 42.3% “intermediate” and 6.7% were self considered as “advanced”. It was intended as an introductory course on fundamentals of magnetic resonance imaging and acquisition. The students were provided with several videotutorials on MR background, where both technological and clinical perspectives were provided; instructors had either an engineering or a radiological background. The students also had access to several simple interactive in-house web applications that graphically illustrated additional concepts (resolution as a function of k-space excursion, k space filling order and translation into image space, to mention two of them). Then, the students were provided with some exercises to be accomplished with the simulator. Each week concluded with a test to monitor student progress.
2. On-line course in 2019: Similar characteristics as the 2018 course, although the estimated dedication increased to 6-7 hours per week to give more emphasis to

the simulator role; this decision was made after reading students suggestions at the end of the 2018 course (the most relevant of which are shown in Table 2). The number of students was 180. Professional profile percentages were now 46% radiologists, 45.5% radiographers. About computer skills, 42% answered “basic”, 51% “intermediate” and 7% were self considered as “advanced”. In this course, the ITS was provided as an optional tool and those who made use of it were kindly asked to fill a similar questionnaire to that given in the previous experience. The course edition was approved by the Spanish Medical Professional Accreditation Council for CPD/CME¹. integrated in the European Accreditation Council for CME. We estimate that the simulator was used at least for one third of the overall expected dedication time to the course.

3.3.2. *Measurement instrument*

The measurement instrument used to account for user experience in the two experiments consists of three elements, namely:

- **SUS:** This measure has become nowadays one of the most popular questionnaires to evaluate usability [48]. According to [49], the test consists of ten questions where each has five possible answers, ranging from “totally agree” to “totally disagree”. The maximum score achievable is 100. In addition, two subscales have been identified [50], so measures of learnability (questions 4 and 10) and usability (the rest of the 10 questions) can be obtained.
- **Perceived utility:** the questionnaire consists of two questions, namely: Q1) “Has the simulator been useful to learn about the principles of magnetic resonance?”; Q2) “Has the simulator been useful to learn about magnetic resonance acquisition?”. The same five answers as before could be selected.
- **LTR:** likelihood to recommend, a score obtained after the question: “Would you recommend other potentially interested people to use our simulator to learn about

¹CPD: continuous professional development; CME: continuous medical education

the principles and acquisition of magnetic resonance? The same five answers as in SUS could be responded and, therefore, the maximum score would be 5 points.

As for the 2019 course, questions were referred to the joint use of the simulator and the ITS. In both experiences, students were informed that they could choose whether or not to take part of the experiments and all the results that will be provided in subsequent sections come from students who explicitly volunteered to being part of our study. Information was anonymized before any data analysis procedure took place.

4. Results

116 (65%) responses were collected from the first on line course and 27 (15%) from the second experience; we should remind that for the second experience only those who volunteered both to use the ITS and to answer the questions have been reported; this is the reason why the number of collected responses is considerably lower. In any case, SUS results are considered stable if the number of participants exceeds 12 [51], which is the case for the two experiments.

SUS and LTR results are shown in Table 1. Perceived utility is represented in Figure 7.

Table 1: Results of SUS and LTR.

		2018 on line course	2019 on line course
SUS	Global	55.86	68.33
	Usability	59.78	68.75
	Learnability	40.19	66.66
LTR		3.65	4.15

We have also collected the most relevant comments provided by the students in free text format. Table 2 shows this student feedback.

5. Discussion

From the first column of Table 1, SUS results are considered as “OK” according to [52]. If we now concentrate on the second column of the table—recall that this column incorporates CTAT—SUS results provide a non negligible increase, and we are now on

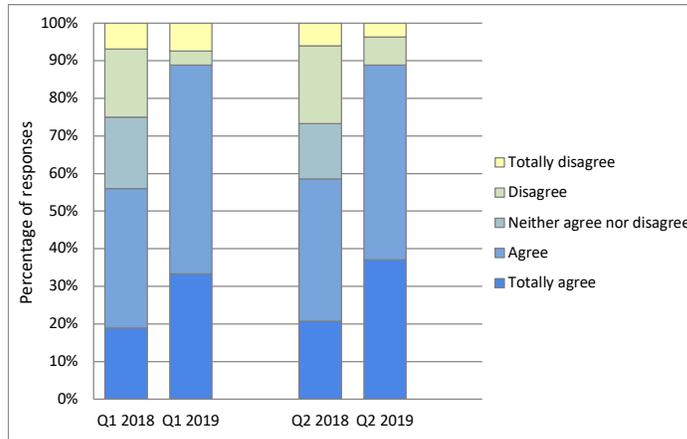


Figure 7: Results of Perceived utility.

the threshold for category “Good” [52]. This improvement seems quite meaningful if we bear in mind the inherent complexity that the MR equipment has; an MR simulator cannot artificially simplify the GUI to make it more usable than actual consoles are, but our results show that the inclusion of an ITS makes the joint system more usable.

If we now look deeper into the two subscales, learnability is lower than usability, which suggests that learning how to use the tool is a limiting factor in the overall SUS. But, as indicated, if learning of MR acquisition was simple, no tools would be needed. We should stress, however, that the incorporation of the ITS gives rise to a clear increase in learnability, which is quite a positive effect since the simulator has been designed to closely resemble an actual MR console.

Results of perceived utility show an obvious improvement as well; specifically, the bars in Figure 7 show that 90% of the students have provided an answer in the two higher most categories for Q1 and Q2 in the 2019 course while the bars in the 2018 course do not reach 60%. Therefore, students have a feeling that the simulator with the ITS has had a positive impact in their learning experience. This seems coherent with the findings reported elsewhere about the benefits of using ITSs in learning experiments (recall Section 2).

The improvements we have observed in the figures of usability, learnability and perceived utility give rise to an increase as well in the LTR. Those quantitative results

2018 course edition		
	Strengths	Weaknesses
1	The simulator is a very valuable toolthat has not been correctly integrated in the course.
2		I would add more exercises to be accomplished with the simulator
3		There should be a higher number of exercises to be solved with the simulator.
4		I would be very interesting to integrate more the simulator with the explained concepts.
5	The simulator is a great tool, nice achievement.	
6		In my opinion [the course] should focus on image acquisition, the right way to do it; good geometric planning and many more exercises with the simulator (which was not easy to use since I did not get any acquisition and it was not excessively intuitive).
7		The simulator should be improved since it is a bit slow and that is why I did not use it much, in this part I would put more practice.
2019 course edition		
	Strengths	Weaknesses
1	I would like to keep on using the simulator when the course is over. It is of great help! Thank you.	
2	It is a very practical programbut it takes a number of improvements to be more efficient when it comes to provide suggestions about exercise execution.
3	I think it is something incredible and wonderful. When I was a student I wish I had had a tool like the simulator and the ITS	
4	I think it is very useful to put in practice the acquired knowledge. I think this tool is of great value for learning.	It would be good to have more cases of this type at disposal.
5	The simulator is great, I never thought something like this could be done.	But I miss somebody that I comment with whether I am doing correctly or to solve doubts. I am sure there is, but I am not very familiar with new technologies and it is hard.

Table 2: Student comments in free text format. Translation into English of the original message set *as is* was carried out.

seem perfectly aligned with the comments provided by the students in Table 2; a simple overview reflects many more entries in the *Strengths* column in the lower half of the table (reflecting comments from the 2019 course) with respect to those in the upper half (from the 2018 course).

This being said, we also understand that the comments in the table suggest different ways for improvement, both directly related to the simulator and how the simulator is used in the course.

Overall, our evidence indicates a feeling of satisfaction with the simulator when an ITS is included, despite further effort has to be done for additional improvement.

6. Conclusions

In this paper we have proposed the inclusion of an ITS in an MR simulator designed and implemented with educational purposes. We have provided details of the simulator itself as well as on how the integration of the two applications can be carried out. In addition, we have compared two actual educational experiences, one of them using the simulator as a standalone instrument and the other one with the ITS integrated within the simulator. Results have provided evidence about the benefits of such an integration in terms of quantitative user experience as well as of qualitative comparative comments directly provided by the students of both courses.

The main limitation of our work is the lack of an educational evaluation of the ITS. We have conducted an evaluation in terms of the benefits provided by the standalone simulator with respect to a more traditional slide-guided presentation [15] and it is expected, based on literature findings in other application domains, that the ITS will further help students; strictly speaking, however, we have not carried out this experiment so far and our evaluation is limited to user experience. Additional work includes this evaluation as well as enlarging the set of exercises for which the ITS will provide; this will let us analyze the typology of exercises that provide higher student satisfaction.

7. Acknowledgements

The authors would like to thank the Spanish Society of Radiology (SERAM) for the opportunity given to the authors to participate in two educational experiences. The authors would also like to thank the Junta de Castilla y León, Spain, and the Fondo Europeo de Desarrollo Regional (FEDER) and by the company Giveme5D for funding this work in part. Grant TEC2017-82408-R should also be acknowledged.

8. Compliance with Ethical Standards

Funding: This work has been funded in part by the Junta de Castilla y León, Spain, and the Fondo Europeo de Desarrollo Regional (FEDER) and by the company Giveme5D, Valladolid, Spain.

Conflict of Interest:

Author Daniel Treceño-Fernández programmed the MRI simulator in cooperation with the “Laboratorio de Procesado de Imagen” at the University of Valladolid and the company Giveme5D.

Author Juan Calabia-del-Campo is the main shareholder of the Giveme5D company.

Author Miguel L. Bote-Lorenzo is a professor at the University of Valladolid.

Author Eduardo Gómez-Sánchez is a professor at the University of Valladolid.

Author Rodrigo de Luis-García is a professor at the University of Valladolid and cooperates with the Giveme5D company in compliance with the Spanish regulation.

Author Carlos Alberola-López is a professor at the University of Valladolid and cooperates with the Giveme5D company in compliance with the Spanish regulation .

Informed consent: Informed consent was obtained from all individual participants included in the study.

References

- [1] G. Torheim, P. A. Rinck, R. A. Jones, and J. Kvaerness, “A simulator for teaching MR image contrast behavior,” *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 2, pp. 515–522, 12 1994.
- [2] T. H. Jochimsen and M. von Mengershausen, “ODIN: Object-oriented development interface for NMR,” *Journal of Magnetic Resonance*, vol. 170, pp. 67–78, 10 2004.
- [3] T. Hackländer and H. Mertens, “Virtual MRI: A PC-based simulation of a clinical MR scanner,” *Academic Radiology*, vol. 12, pp. 85–96, 02 2005.
- [4] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, and C. L. Odet, “The SIMRI project: a versatile and interactive MRI simulator,” *Journal of Magnetic Resonance*, vol. 173, no. 1, pp. 97 – 115, 2005.
- [5] I. Drobnjak, D. Gavaghan, E. Süli, J. Pitt-Francis, and M. Jenkinson, “Development of a functional magnetic resonance imaging simulator for modeling realistic

- rigid-body motion artifacts,” *Magnetic Resonance in Medicine*, vol. 56, pp. 364–80, 08 2006.
- [6] I. Drobnjak, P. G. S., and M. Jenkinson, “Simulating the effects of time-varying magnetic fields with a realistic simulated scanner,” *Magnetic Resonance Imaging*, vol. 28, no. 7, pp. 1014 – 1021, 2010.
- [7] G. Andria, F. Attivissimo, G. Cavone, and A. M. L. Lanzolla, “Acquisition times in magnetic resonance imaging: Optimization in clinical use,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 3140–3148, 9 2009.
- [8] T. Stöcker, K. Vahedipour, D. Pflugfelder, and N. J. Shah, “High-performance computing mri simulations,” *Magnetic resonance in medicine*, vol. 64, no. 1, pp. 186–193, 2010.
- [9] C. G. Xanthis, I. E. Venetis, and A. H. Aletras, “High performance mri simulations of motion on multi-gpu systems,” *Journal of Cardiovascular Magnetic Resonance*, vol. 16, p. 48, 7 2014.
- [10] C. G. Xanthis, oannis E. Venetis, A. Chalkias, and A. Aletras, “MRISIMUL: A GPU-Based Parallel Approach to MRI Simulations,” *IEEE Transactions on Medical Imaging*, vol. 33, pp. 607–617, 2014.
- [11] M. S. Graham, I. Drobnjak, and H. Zhang, “Realistic simulation of artefacts in diffusion mri for validating post-processing correction techniques,” *NeuroImage*, vol. 125, pp. 1079–1094, 2016.
- [12] M. S. Graham, I. Drobnjak, M. Jenkinson, and H. Zhang, “Quantitative assessment of the susceptibility artefact and its interaction with motion in diffusion mri,” *PloS one*, vol. 12, no. 10, p. e0185647, 2017.
- [13] R. Kose and K. Kose, “Blochsolver: A gpu-optimized fast 3d mri simulator for experimentally compatible pulse sequences,” *Journal of Magnetic Resonance*, vol. 281, pp. 51–65, 2017.

- [14] D. Treceño-Fernández, J. Calabia-del-Campo, A. Luna-Alcalá, M. Bote-Lorenzo, E. Gómez-Sánchez, R. de Luis-García, and C. Alberola-López, “A Web-based Educational Magnetic Resonance Simulator: Design, Implementation and Testing,” -, *Journal of Medical Systems* (in press).
- [15] D. Treceño-Fernández, J. Calabia-del-Campo, M. Bote-Lorenzo, E. Gómez-Sánchez, R. de Luis-García, and C. Alberola-López, “A web-based MRI simulator for radiographer education: Quantitative evaluation of an actual classroom experience,” in -, *Annual Conference of Radiological Society of North America (RSNA)*, Chicago, USA, 2019.
- [16] J. R. Carbonell, “Ai in cai: An artificial-intelligence approach to computer-assisted instruction,” *Man-Machine Systems, IEEE Transactions on*, vol. 11, pp. 190 – 202, 01 1971.
- [17] D. Sleeman and R. J. Brown, *Intelligent tutoring systems*. Academic Press, 1982.
- [18] R. W. Hill Jr, J. Belanich, H. C. Lane, M. Core, M. Dixon, E. Forbell, J. Kim, and J. Hart, “Pedagogically structured game-based training: Development of the elect bilat simulation,” in *Proceedings of the 25th Army Science Conference*, Proceedings of the 25th Army Science Conference, 2006.
- [19] S. Mulgund, M. Asdigha, G. Zacharias, K. Krishnakumar, and J. Dohme, “Intelligent tutoring system for simulator-based helicopter flight training,” in *Flight Simulation Technologies Conference*, 10 1998.
- [20] R. H. Stottler and M. Vinkavich, “Tactical action officer intelligent tutoring system (tao its),” in *”Training, Simulation and Education Conference (IITSEC 2000)”*, Proceedings of IITSEC, 2000.
- [21] J. Mostow and G. Aist, “Evaluating tutors that listen: An overview of project listen.,” in *Smart machines in education: the coming revolution in educational technology*, pp. 169–234, MIT Press, 2001.

- [22] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam, “Effectiveness of cognitive tutor algebra i at scale,” *Educational Evaluation and Policy Analysis*, vol. 36, no. 2, pp. 127–144, 2014.
- [23] I. Arroyo, B. P. Woolf, W. Burelson, K. Muldner, D. Rai, and M. Tai, “A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect,” *International Journal of Artificial Intelligence in Education*, vol. 24, pp. 387–426, Dec 2014.
- [24] K. G. Schulze, R. N. Shelby, D. J. Treacy, M. C. Wintersgill, K. VanLehn, and A. Gertner, “Andes: An intelligent tutor for classical physics,” *Journal of Electronic Publishing*, vol. 6, 09 2000.
- [25] J. Sabourin, L. Shores, B. Mott, and J. C. Lester, “Understanding and predicting student self-regulated learning strategies in game-based learning environments,” *International Journal of Artificial Intelligence in Education*, vol. 23, 06 2012.
- [26] E. Mousavinasab, N. Zarifsanaiey, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saeedi, “Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods,” *Interactive Learning Environments*, pp. 1–22, 2018.
- [27] K. VanLehn, “The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems,” *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011.
- [28] W. Ma, O. O. Adesope, J. C. Nesbit, and Q. Liu, “Intelligent tutoring systems and learning outcomes: A meta-analysis.,” *Journal of educational psychology*, vol. 106, no. 4, p. 901, 2014.
- [29] S. S. Jeon, G. Lee, S. Lamptang, and S. Y. Su, “An integrated e-learning system for simulation-based instruction of anaesthesia machines,” *International Journal of Knowledge and Learning*, vol. 3, no. 1, pp. 106–120, 2007.

- [30] S. S. Jeon and S. Y. Su, “Adaptive e-learning using ecpaa rules, bayesian models, and group profile and performance data,” *International Journal of Learning Technology*, vol. 5, no. 4, pp. 415–434, 2010.
- [31] H. Kazi, P. Haddawy, and S. Suebnukarn, “Employing umls for generating hints in a tutoring system for medical problem-based learning,” *Journal of biomedical informatics*, vol. 45, no. 3, pp. 557–565, 2012.
- [32] P. Xiberta and I. Boada, “A new e-learning platform for radiology education (RadEd),” *Computer methods and programs in biomedicine*, vol. 126, pp. 63–75, 2016.
- [33] K. Almohammadi, H. Hagra, D. Alghazzawi, and G. Aldabbagh, “Users-centric adaptive learning system based on interval type-2 fuzzy logic for massively crowded e-learning platforms,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 6, no. 2, pp. 81–101, 2016.
- [34] M. A. Bernstein, K. F. King, and X. J. Zhou, eds., *Handbook of MRI Pulse Sequences*. Burlington: Academic Press, 2004.
- [35] C. Burch, “Django: a web framework using python: Tutorial presentation,” *Journal of Computing Sciences in Colleges*, vol. 25, pp. 154–155, 5 2010.
- [36] C. Anderson, *The Model-View-ViewModel (MVVM) Design Pattern*, pp. 461–499. Berkeley, CA: Apress, 2012.
- [37] B. P. Woolf, *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.
- [38] T. Murray, “An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art,” in *Authoring tools for advanced technology learning environments*, pp. 491–544, Springer, 2003.
- [39] A. Mitrovic, B. Martin, P. Suraweera, K. Zakharov, N. Milik, J. Holland, and N. McGuigan, “Aspire: an authoring system and deployment environment for

- constraint-based tutors,” *International Journal of Artificial Intelligence in Education*, vol. 19, no. 2, pp. 155–188, 2009.
- [40] V. Aleven, B. M. McLaren, J. Sewall, M. Van Velsen, O. Popescu, S. Demi, M. Ringenberg, and K. R. Koedinger, “Example-tracing tutors: Intelligent tutor development for non-programmers,” *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 224–269, 2016.
- [41] S. B. Gilbert, S. B. Blessing, and E. Guo, “Authoring effective embedded tutors: An overview of the extensible problem specific tutor (xpst) system,” *International Journal of Artificial Intelligence in Education*, vol. 25, no. 3, pp. 428–454, 2015.
- [42] L. Paquette, J.-F. Lebeau, G. Beaulieu, and A. Mayers, “Designing a knowledge representation approach for the generation of pedagogical interventions by mmts,” *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 118–156, 2015.
- [43] A. C. Graesser, “Conversations with autotutor help students learn,” *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 124–132, 2016.
- [44] H. C. Lane, M. G. Core, M. J. Hays, D. Auerbach, and M. Rosenberg, “Situated pedagogical authoring: Authoring intelligent tutors from a student’s perspective,” in *International Conference on Artificial Intelligence in Education*, pp. 195–204, Springer, 2015.
- [45] R. Sottolare *et al.*, “Considerations in the development of an ontology for a generalized intelligent framework for tutoring,” in *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*, pp. 19–25, 2012.
- [46] K. R. Koedinger and V. Aleven, “Exploring the assistance dilemma in experiments with cognitive tutors,” *Educational Psychology Review*, vol. 19, no. 3, pp. 239–264, 2007.
- [47] V. Aleven, B. M. McLaren, and J. Sewall, “Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site

for middle school mathematics learning,” *IEEE Transactions on Learning Technologies*, vol. 2, no. 2, pp. 64–78, 2009.

- [48] J. Brooke, “Sus: A retrospective,” *J. Usability Studies*, vol. 8, pp. 29–40, Feb. 2013.
- [49] J. Brooke, “Sus: A quick and dirty usability scale,” *Usability Eval. Ind.*, vol. 189, 11 1995.
- [50] J. R. Lewis and J. Sauro, “The factor structure of the system usability scale,” in *Human Centered Design* (M. Kurosu, ed.), (Berlin, Heidelberg), pp. 94–103, Springer Berlin Heidelberg, 2009.
- [51] T. S. Tullis and J. N. Stetson, “A comparison of questionnaires for assessing website usability,” in *Usability Professionals Association (UPA) 2004 Conference*, Jun 2004.
- [52] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *J. Usability Studies*, vol. 4, pp. 114–123, 5 2009.

Supplementary Materials

With the technology we have used, AngularJS, the two actions enumerated in Section 3.2.3 are easily done by services and directives, respectively. With respect to the service, the following piece of code creates a service named `dtCTATTools`,

```
1 service('dtCTATTools', ['$rootScope', 'coursesData', function($rootScope,
2     coursesData){
3     //Main variables
4     //-----
5     var data = {};
6     data.CTATEnabled = false;
7     data.CTATMyVars = {};
8     data.myProblem = 1;
9     data.myCourse = 1;
10    data.graphConfig = null;
```

```

11 data.problemConfig = null;
12 data.recordAssociatedRules = {
13     flag:true/false,
14     recorded: [],
15 };
16
17 //Main functionality
18 //-----
19 var dtCTATToolsPointer = {};
20 dtCTATToolsPointer.setCTATAuthoring = function(){ ... };
21
22 //...
23 });

```

The variables contained within the object data (line 5) are only accessible through the object dtCTATToolsPointer (line 19); we provide an example of a setter function associated to this object (line 20). As for the variables within data, they can be defined at any time. For instance, the variable CTATEnabled (line 6) is defined at the beginning of the Web page load. The variable CTATMyVars (line 7) contains all the parameters that are needed to configure CTAT and it is defined right before CTAT is initialized.

With respect to the directives, we will use one of them as an example; it is called dtCtatSelect and we have included a piece of code that illustrates its rationale.

```

1 .directive('dtCtatSelect', ['dtCTATTools', function(dtCTATTools){
2     ...
3     var dtDirectiveSelect = {}; //
4     var dtCTATSelect = function(){
5         CTAT.Component.Base.Clickable.call(this, "dtCTATSelect", "dtCTATSelect");
6         ...
7         this.init=function init(){
8             var _id = this.getName();
9             var select = dtDirectiveSelect[_id].returnSelect();
10            this.setComponent(select);
11            this.newChoice = dtDirectiveSelect[_id].newChoice;
12            ...
13        };
14    };
15
16    ...
17    function main(scope, element, attr, ctrl){
18        window.setTimeout(onLoad(scope, element, attr, ctrl), 0);
19        function onLoad(scope,element,attr,ctrl){
20            if(!(String(attr.id) in dtDirectiveSelect)){

```

```

21     Object.defineProperty(dtDirectiveSelect, attr.id, {value:{}});
22     }
23     dtDirectiveSelect[attr.id].returnSelect = function(){...};
24     dtDirectiveSelect[attr.id].newChoice = function() {...};
25     ...
26     }
27 }
28 return {
29     restrict: "A",
30     require: "ngModel",
31     link: main,
32 };
33 })

```

The directive starts by injecting the service previously defined (line 1). This directive is used to create a constructor of a component that inherits from a CTAT component; in this example, the component is of type `Clickable` (line 5). The key of the directive, however, is the object `dtDirectiveSelect` (line 3), which will be accessed even from within the functions that CTAT executes.

The function `init` (line 7) declares two methods, namely, `returnSelect` (line 9) and `newChoice` (line 11) which are empty when CTAT is loaded. These two methods, however, are associated to each component of type `Select` which are specified by a unique `id`. The function `main()` (line 17) is executed once for each directive declared, including those that are dynamically generated. When the function is executed, a new element is added to `dtDirectiveSelect` according to its `id`. In addition, the two methods that were declared are now properly defined (lines 23 and 24) and will consist of the functionality that the programmer specifies. With this methodology, CTAT is able to register all the components of the type `Select`, independently of the moment in which they are created. Dynamics objects are, consequently, dealt with naturally, so we do not need to resort to any *ad-hoc* solution for dynamic objects.