Improving Decision-making in Virtual Learning Environments using a Tracing Tutor Agent

Aluizio Haendchen Filho[1,2], Jeferson Miguel Thalheimer[1], Rudimar L. S. Dazzi[1], Vinícius Santos[1] and Paulo Ivo Koehntopp[3]

[1]*Laboratory of Applied Intelligence, University of the Itajaí Valley (UNIVALI), Rua Uruguay, 458, Itajaí, Brazil*
[2]*Univertity Center of Brusque (UNIFEBE), Brusque, Brazil*
[3]*Catarinense Association of Educational Foundations (ACAFE), Florianópolis, Brazil*

Keywords: Intelligent Tutor, Software Agents, Decision-making, Virtual Learning Environment.

Abstract: Quality of care in the Virtual Learning Environment is often compromised by large numbers of students. This presents a difficult task for human tutors. On the other hand, Intelligent Tutoring Systems are evolving towards a decision support system. One vision of Artificial Intelligence and education is to produce a tutor for every student or a "community of tutors for every student". Here we present a model of intelligent tracing tutor agent responsible for tracking students in the virtual learning environment. We have designed the Tracing Tutor Agent as one of the agents of a collaborative organization of intelligent tutor agents. Each agent has his role, responsibilities and permissions. The main focus of this work is to present a model of Tracing Tutor Agent (TTA), which is one of the organization's agents. It has the following responsibilities: (i) monitor students' actions in VLE; (ii) to monitor the actions of the human tutor in the VLE; and (iii) collaborate and interact with the other organization's agents to supply the human tutor with information in order to improve decision making and performance, increasing attendance and avoiding evasion.

1 INTRODUCTION

The possibility of investing in universities and private courses have made the educational market very attractive and profitable for national and international groups that have been constantly involved in large, million-dollars negotiations. With the technological advancements and the spread of distance learning courses, the number of students entering this environment has significantly increased every year.

Despite all the technological advances and concerns to ensure courses quality, Virtual Learning Environments (VLE) continue to present significant problems (Tomelin, 2016). The quality of care in the VLE is often compromised by large numbers of students. Consequently, problems including demotivation and sentiments of isolation arise, causing low use and high evasion rate.

According to Vicari and Giraffa (2003), with the emergence of Artificial Intelligence (AI), the developers began to use Computed Assisted Instruction (CAI) techniques to make the software more active in the process of student interaction. From that initial experience emerged the Intelligent Computer-assisted Instruction (ICAI*)* method and following that Intelligent Tutoring Systems (ITSs). Shifting from a problematic Expert System to an ITS appeared to be a natural course.

In the search for automation and trying to reproduce the teacher pedagogy, the tutors were initially developed with the purpose of replacing the teacher. With the evolution of virtual learning environments and the emergence of active learning methodologies, ITSs are evolving towards a decision support system. One vision of Artificial Intelligence and education is to produce a "tutor for every student", or a "community of tutors for every student." This vision includes the process of learning in social activity, accepting multimodal input from students (Woolf, 2010). Thus, the role of intelligent tutors needs to be adapted to this new reality.

We have designed the Tracing Tutor Agent as a collaborative agent of an intelligent tutor from the organization. Each agent has his own role, responsibilities and permissions. Many of the agents' activities are executed in real time. When their lifecycle starts, they remain in constant monitoring of the virtual learning environment. In order to properly fulfill their

Filho, A., Thalheimer, J., Dazzi, R., Santos, V. and Koehntopp, P.
Improving Decision-making in Virtual Learning Environments using a Tracing Tutor Agent.
DOI: 10.5220/0007744006000607
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 600-607
ISBN: 978-989-758-372-8

roles, the collaboration and interaction between them are two fundamental properties.

The main focus of this work is to present a model of Tracing Tutor Agent (TTA). It has the following responsibilities: (i) monitor the actions of students in VLE; (ii) monitor the actions of the human tutor in the VLE; and (iii) collaborate and interact with the other organization's agents to supply the human tutor with information in order to improve decision making and performance, increasing attendance and avoiding evasion.

2 BACKGROUND

To substantiate the approach, this section presents the following concepts: (i) intelligent tutors; (ii) data webhouse and (iii) agent development platform.

2.1 Intelligent Tutors

Traditionally, ITSs are educational computational systems that aim to promote immediate and customized instruction to provide students with individualized learning (Vos, 1995, Van Lehn, 2006, Bernacki et al., 2014, Latham et al. Lin et al., 2014). These instructions are performed without human intervention, making the system simulate the behaviour of a human tutor. In this way, tips are introduced to help the student to correctly develop the problem and identify errors (Wenger, 1987; Azevedo et al., 1999; Curilem (2007), and Adams et al. (2014).

ITSs have been investigated by several authors, including Raabe (2005), Giraffa (1999), San Pedro, Baker and Rodrigo (2014), Rissoli, et al. (2006), and more recently Mayer et al. (2014).

According to Rissoli, Giraffa and Martins (2006) ITSs have the ability to learn and teach. In this way, ITSs allow adaptation of teaching strategies to the learning of each student's needs. The dynamic and coherent combination of information and the content of pedagogical aspects are related to every student.

According to Dazzi (2007), ITSs are computer systems that tutor a student in a given domain. ITSs models the students' understanding of a topic. As it performs certain tasks in the system, it compares the student's knowledge to the model they have of an expert in that domain. If there is a difference, the system can use its domain model to generate an explanation to help the student to identify the mistake. The system can also adjust student learning levels and styles and present information, tests and more appropriate feedback.

Ausubel (1980) proposes the Significant Learning Theory (SLT) method, which helps to clarify the construction of new knowledge based on the learner's cognitive structure. In the educational context, this theory promotes respect for the characteristics of students. This demands more from their own educationnal processes rather than their sociocultural particularities. It requires a personalized and interactive education, which allows the advancement of each student considering their particular needs. This personalized education is not feasible in traditional teaching and learning, where a large number of students are under the pedagogical responsibility of a single tutor (Souza et al., 2013).

Research conducted by Azevedo et al. (2016) and Mudrick, et al. (2017) support the idea that assistance and feedback of virtual pedagogical agents' favour self-regulated and complex student learning. This depends on tracking, modelling and incentive to the intelligent and precise feedback (apud Barcenes et al., 2018). Other authors like (Grawemeyer et al., 2017) developed a system to improve commitment and learning through formative feedback based on the emotional state of the student, called iTalk2Learn.

In this sense, we observe the great potential of the systems that offer personalized education and tracking of students' actions.

2.2 Data Webhouse

According to Kimball and Merz (2000), the Web allows the possibility of recording practically all behavioural actions of the user in a single click. In terms of behavioural actions, it must be understood that one can capture not only the page accessed but also weather and navigability information.

To make their proposal feasible a technique called clickstream is used for the exploitation of Web access information. The recording of all interactions made by anyone via an application or web site, is literally called a clickstream. Activities carried out by the user such as click capturing, form filling, and others, create conditions for analysis, profile identifications, preferences and trends of each particular user.

Figure 1 shows a very simple example of a dimensional model for DWH.

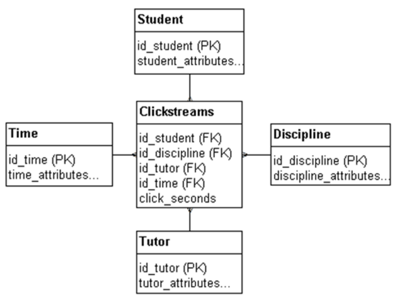

Figure 1: Simple dimensional DWH example.

As for Data Warehouse, Data Webhouse (DWH) is built with an architecture called Dimensional Model. Dimensional modelling is a discipline that seeks to model data for the purposes of understandability and performance.

All dimensional models are built around the concept of measured facts. The Facts table, represented by the entity Clickstreams, stores users' clicks on the VLE. The dimensions relate to the entities that serve as perspectives of analysis in any subject of the model. In the example, Student, Discipline, Time and Tutor are the dimensions connected to the fact table. Dimensions are rich in descriptions. For example, the Student dimension stores all the student profile data.

2.3 Agent Development Platform

The inherent difficulties of agent-based systems development induce the search for tools in order to minimize complexity and reduce time and cost. For the development of the solution, we use the Midas Platform (Haendchen Filho, Prado, and Lucena, 2007). Midas offers a flexible, extensible, adaptive, loosely coupled, and service-oriented architecture in order to interoperate in the Web. The platform provides a complete runtime environment where agents can execute concurrently within the same host.

Figure 2 shows the Midas generic architecture detailing the Agent Container (AC). AC is a web container where application agents and components are hosted, providing a platform and a framework to simplify the Multi Agent System development. The mechanisms embedded in Midas work in two abstraction levels: (i) the generic architectural level, where four middleware agents are responsible for providing infrastructure services; and (ii) in the agents' design level, providing abstract classes that define hot-spots from which concrete agents and/or components can be developed. The abstract template already empowers the agents with communication facilities, including event-based listeners and a blackboard.

The reference architecture is composed of five models: (i) Message-Oriented Model (MOM): dedicated to the architectural issues about the structure and transport of messages; (ii) Service-Oriented Model (SOM): focussed in the services and actions executed by requesters and providers; (iii) Resource-Oriented Model (ROM): focussed in the architectural aspects about the resources handling; and (iv) Management Model (MGM): devoted to the management of resources.

Blackboard is one of the most used information exchange techniques in symbolic cognitive Multi-Agent System. Its structure follows the basic blackboard pattern: the knowledge sources represent the agents, the data structure is visible to all agents, and the controller is responsible for notifying the agents about the changes in the environment.

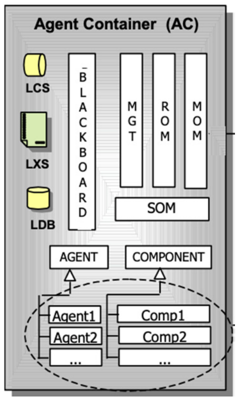

Figure 2: Agent Platform (Haendchen Filho et al. 2007).

Agents are autonomous entities that have their own thread execution and can implement adaptive or intelligent behaviour. The abstract class provides the interfaces and triggers the agent execution thread and the signatures for lifecycle management. The component is an abstract class that represents purely reactive entities, typically used to encapsulate the domain-specific rules of the application, including business processes, data access objects, and legacy application functionalities.

3 PROPOSED APPROACH

This section is intended to present the main specifications of the system. The structure is divided in (i) Generic Model; (ii) Learning Model; (iii) VLE Webhouse Model; and (iv) Tracing Tutor Agent Role, presented below.

3.1 Generic Model

The generic model is composed of an organization of intelligent tutors, data sources, VLE and two main actors: the human tutor and the student, as shown in Figure 3.

Intelligent Tutor System is a software agent organization composed of the following agents: (i) TTA (Tracing Tutor Agent), which represents the tracer tutor, which is the main object of this paper; (ii) ITA (Interface Tutor Agent), which represents an interface tutor; its role is to provide information and notification adaptive graphical interfaces in VLE; (iii) PTA (Pedagogical Tutor Agent), that follows students' interaction with the VLE, collect the information required for the modelling of students' profile used to customise the environment assist and guide students during the construction of their learning (Dos Santos et al., 2002); (iv) AITA (Artificial Intelligence Tutor Agent), which manipulates artificial intelligence techniques like machine learning and data mining techniques that perform predictions and prescriptions; (v) Student representing a virtual student agent, (vi) Tutor representing a virtual Tutor, and (vii) Component, representing reactive objects used by agents to obtain information stored in the database. The databases are represented in the figure by the DWH and Academic Information System (AIS).

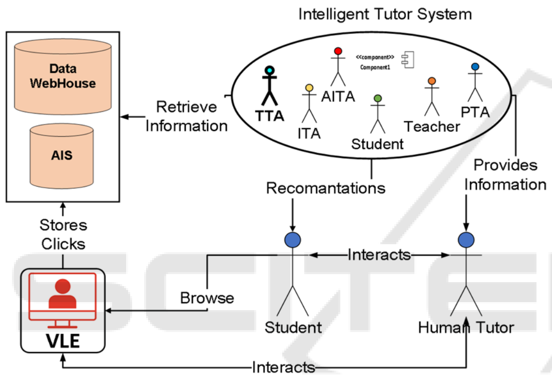

Figure 3: Generic model.

The student accesses the VLE, having its clicks captured and stored in a Data Webhouse during its navigation. After obtaining the data and performing the analyses, the TTA provides the information to the human tutor. With this information, the human tutor can make decisions and send notifications or keep in touch with students detected on alert. Also, the TTA may obtain permissions from the human tutor to send notifications and help students with difficulty.

In VLE, listeners are placed in relevant spot, waiting for the clicks to trigger the script to store the information. Locations do not necessarily have to be on the links since with dynamic page load, data can be stored with simple interactions. Clickstreams, about the click data to the user ideally, should be sent to the webhouse data for storage.

All actions the user take can disclose knowledge about the use of the system. Obtain such knowledge the data webhouse is widely used to process analyses, obtaining information from two main sources: (i) communication protocol data, stored in the web services logs; and (ii) behaviour seized with site scripts after establishing a session. User's behaviour on pages is a critical part because it is not so simple to change a site to capture the information.

There are two approaches to deploy scripts on site pages: (i) placing a script within each existing clickable element; and (ii) create a script to listen to data on all pages and filter the relevant ones. By placing a script on the clickable elements, it is possible to ensure that all important information is correctly captured. The disadvantage of this approach is to generate much work on site maintenance. On the other hand, using a script to listen to all page data is a simpler approach, but it has the disadvantage of losing relevant information about what is being clicked on.

It is important to clarify that a significant part of student clicks, especially internal clicks in VLE, can be collected in the log files generated by some platforms, such as Moodle (Dougiamas, 2001) and others. In addition to the clicks performed on the VLE, the system may also capture clicks from other browser open windows during the study. To enable this, the system will prompt users for permission to use cookies on their computer. With cookies, it is possible to capture various preferences and interests of users, and especially the external pages visited during the learning process, such as collecting articles and materials related to the topics of the classes.

3.2 Domain Learning Model

VLEs usually have a model called the domain model, where the contents practices and learning pathways are defined. In the domain model, the schedules are defined to allow the contents' presentation to the students in the diverse pedagogical strategies of the system. Figure 4 adapted from (Positivo, 2017) presents a learning pathway model. The example shows the activities of a weekly learning schedule.

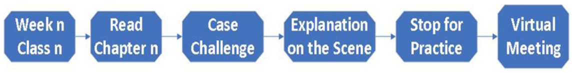

Figure 4: Model of learning pathway.

In the learning process the student navigates in the VLE looking to follow the pathways defined in the domain model. During navigation, the student performs several actions, such as the reading of teaching material, participate in challenge cases, attends video conference explanation, performs the practical activities requested by the teacher, and participates in virtual meetings like forums, chats or others. One of the roles of TTA is to monitor students' actions in the learning pathway, as will be shown in the role model.

3.3 VLE Webhouse Model

As presented in Section 2, in a Data Webhouse (DWH), data is stored in dimensional models. Dimensional modelling can begin by dividing it into two parts: measurement (facts) and context (dimensions). Measurements are usually numeric and repeated. Facts are usually surrounded by a large textual context (dimensions). Figure 5 shows the DWH dimensional model.

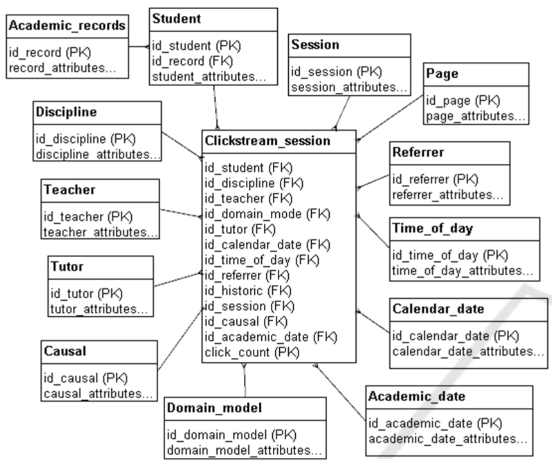

Figure 5: VLE Webhouse Model.

The template contains a central Fact table connected with the dimensions relevant to the context of the VLE. The relationships among Facts table and dimensions are made by their primary keys.

Users clickstreams and/or collected logs are stored in the Fact Table, represented by the *click_count* attribute. This attribute is an array type, and contains the data collected in the clicks or log files, such as the page id, the start time (second / minute / hour / month / day / year) of the event, event name, description of what happened in the event, the previous source or page, the user's *IP* address, and so on.

The summary descriptions of dimensions are given below:

- *Calendar date*: attributes may include days of the week, seasons, holiday, workday, tax period, among others.
- *Time_of_day*: is a complement to the calendar date, where are recorded the time slots during the day, including hours, minutes, and time slots like lunchtime, class time, and so forth.
- *Academic date*: associated with different structures that differ on the number of modules (semesters, four-month periods and trimesters). These structures have a different hierarchy, each with five levels: Year, Module (6, 4 or 3 months), Period (classes or examination period), Week (variable length, defined internally by each institution) and Day.
- *Page*: among the attributes of this dimension can be cited: the page source (e.g., static, dynamic), function (content, exercise, video, forum) and so forth.
- *Session*: session is the collection of actions taken by a visitor to a site while it navigates through it without leaving this site.
- *Causal*: describes the conditions of the current progress of the subject, such as beginning of the subject, period of tests, etc.
- *Student:* information about the student profile.
- *Academic records:* provides information on the student's trajectory, like courses taken, historical grading, and so forth.
- *Discipline:* the attributes can include class hours, credits, opening and closing dates, and so forth.
- *Teacher:* information about teacher's profile.
- *Tutor:* information about the tutor including the degree of training, number of tutored disciplines, etc.
- *Domain_model:* describes the course schedule of the learning path in the VLE.
- *Referrer:* brings information about the URL from where the user came from.

In the dimensional model, the student's historical dimension is connected to the student dimension, characterizing a variant of the dimensional model named snowflakes. The main source of information for obtaining dimension data is the institution's Academic Information System.

3.4 Tracing Tutor Role Model

The role model has been used by many approaches (Gonçalves, 2009, Haendchen Filho, 2017) to provide a summary of software agents. According to the theory, a role can be described by two basic attributes: (i) responsibilities: they are role obligations and indicate functionality, and (ii) permissions: they are the rights associated with the role and indicate the resources that the agents can use. Intuitively, responsibilities are associated with services that an agent must provide, while permissions are associated with the services the agent must fulfil their responsibilities.

TTA does not communicate directly with the human tutor or student. Communication tasks with human actors are carried out by ITA. TTA's main responsibilities are to collect data on learning pathways and collaborate with other intelligent tutors (ITA, AITA and PTA).

It provides information for ITA (detailed in the TTA Role Model) that makes it available for VLE managers, human tutors, and students. The data collected in the clicks and/or log files are stored and made available to AITA, which can use this data to feed the machine learning and data mining algorithms. TTA also collaborates with the PTA, which can suggest new pathways and check which trails the students most successfully use. As mentioned, all information for decision making comes to human administrators and tutors via ITA.

TTA's responsibilities are divided into four groups: (i) student monitoring; (ii) tutor monitoring; (iii) Provides Information for ITA and (iv) Statistical Reports for ITA. Figure 6 shows the role model of the agent, with its Responsibilities, Permissions and Collaborations.

ROLE NAME: Tracing Tutor Agent **ORGANIZATION**: Intelligent Tutor System	
RESPONSIBILITIES	**PERMISSIONS**
Service Group: Student Monitoring	**Data Webhose Clickstreams**
Time spent on VLE	Fact table data
Chapter download	Dimensions tables data
Time spent on case challenge	
Time spent on explanation on the scene	**COLLABORATIONS**
Time spent on practice	Interface Tutor Agent (ITA)
Time spent on virtual meeting	Pedagogical Tutor Agent (PTA)
Service Group: Tutor Monitoring	AI Tutor Agent (AITA)
Average response time for doubt	
Time spent on VLE	
Average number of feedbacks	
Service Group: Provides Information for ITA	
Students with delayed tasks	
Students with no attendance at VLE	
Students with little frequency in VLE	
Service Group: Statistical Reports for ITA	
Graphical reports of monitored activities	

Figure 6: TTA Role Model.

The services of the Student Monitoring group comprise tasks related to information about the participation and interactions of the students in the VLE, according to the domain learning model:

- *Time spent by the student on VLE*: identifies students who are not entering the VLE, those who are entering infrequently, and those who are frequent attendees.
- *Chapter download*: verifies if the student has downloaded the materials made available for reading.
- *Time spent on cases challenge*: monitors the amount of time the student spent on the challenges assigned by the teacher.
- *Student time on explanation of the scene*: monitors whether the student has attended and participated in the video conferences.
- *Student time in quiz/tests/exercises*: they are synthesized as Stop for Practice in the learning pathway, verifies if the student is participating in these activities and how much time he spends in each one.
- *Time spent on virtual meetings*: verifies if the student participates in the forums, chats and online activities carried out in the VLE.

The related services in the Tutor Monitoring group comprise activities associated with the activities performed by the tutor in the VLE:

- *Average response time for doubts*: the amount of time that the human tutor or teacher takes on average to answer students' questions.
- *Time spent by the human tutor VLE*: the amount of time the human tutor remains in the VLE.
- *The average number of feedbacks*: counts the average number of feedbacks given to students.

The services listed in the Provides Information for ITA group comprise a sequence of alerts sent from TTA to ITA:

- *Students with delayed tasks*: notify the human tutor about students with overdue assignments.
- *Students with no attendance at VLE*: notify the tutor about students not entering the VLE.
- *Students with little frequency in VLE*: notify the tutor about students entering low on VLE.

The Statistical Reports group represents reports that can be available for ITA with information obtained from the monitoring tasks. For each monitoring service, reports and graphs can be generated to facilitate visualization for analysis and decision-making. With this information, the ITA and the human tutor can give individual attention to the students, taking preventive measures to avoid low achievement or even avoidance.

Role Model Permissions, as well as its Collaborations are placed in the right-side column of Figure 6. As mentioned, the permissions include data and information that the tutor agent can use to fulfil their responsibilities. TTA uses services provided from ITS components in order to obtain data from the database, as well as to generate reports and organize information.

Based on information provided by TTA on interactions, students' course in learning pathways, and assessment, PTA can identify points of difficulty and suggest course corrections in learning paths. This knowledge could hardly be obtained by human tutors in virtual environments with large numbers of students. Moreover, intelligent tutors can be assigned tasks to assistance students in the navigation process, and the best course of learning paths through recommendation systems, optimizing their accomplishment and achievement.

Furthermore, tracking clickstreams will enable the tutor to understand why certain groups to follow a sequence of steps, while others follow different ones. This will allow verification of the track defined in the domain model. The expected quality will hopefully permit continuous improvements in learning pathways and GUIs.

The monitoring of the actions of the human tutor in the VLE allows analysis and self-assessment of their participation in the learning process.

4 DISCUSSION

The approach presented in this paper meets the operational needs requested by the distance learning managers of the University Center of Brusque (UNIFEBE). Specialists in this area point out the importance of information obtained by student tracking. In this proposal, this information is made available by the Tracing Tutor Agent. It provides information for other collaborative agents in order to improve the decision-making of the human tutors and managers in the virtual learning environment.

As before mentioned, Intelligent Tutoring Systems are characterized for incorporating Artificial Intelligence techniques into their design and development, acting as assistants in the teaching-learning process (De Souza et al 2002). Currently, Intelligent Agents concepts have been applied to these systems as a way to improve them.

We chose the Multi-Agent Systems platform to implement our proposal, considering that intelligent software agents act dynamically and not only reactively. They can act in a collaborative way to play their roles more easily. In addition, intelligent agents may learn from the knowledge engendered in the environment, using this learning proactively for the benefit of managers, human tutors and students.

We understand that in an Intelligent Tutors organization each agent will have a specific role model, acting collaboratively. For example, the Interface Tutor Agent can use the information provided by Tracing Tutor Agent to send notifications and create dynamic graphical interfaces, inducing the student to fill gaps in their learning path. PTA collect the information required for the modelling of students' profile used to customize the environment assist and guide students during the construction of their learning. The Artificial Intelligence Tutor Agent (AITA) can identify student clusters that have succeeded in learning. It also provides subsidies for the human tutor and to the managers virtual learning environments, supporting the task of improving the domain model. The AITA may also use machine-learning techniques to identify potential student evasion and prescribe actions to avoid it.

5 CONCLUSION AND FUTURE WORKS

The main contribution of this work is to define a model for intelligent tutors best adapted to the current management needs of virtual learning environments. The focus of this work was to define the role of Tracing Tutor Agent, which is one of the collaborative agents of the organization. Besides we have introduced a Data Webhouse model in the context of intelligent tutor models. We did not find similar works in the literature, which makes the approach innovative.

Future work will involve the implementation of the Tracing Tutor Agent and the development of the specifications of other intelligent tutors presented in the model.

REFERENCES

Adams D. M., McLaren B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., Velsen, M. V., 2014. *Using erroneous examples to improve mathematics learning with a web-based tutoring system Computers in Human Behavior*.

Ausubel, D. P., Novak, J. e Hanesian, H., 1980. *Psicologia educacional.* Rio de Janeiro: Interamericana.

Azevedo, B. F. T., Tavares, O. L., Cury, D., 1999. *Mutantis – Um Sistema Tutor Inteligente Multi-Agente para o Ensino-Aprendizagem de Conceitos.* In: X Simpósio Brasileiro de Informática na Educação, Curitiba. Anais do X Simpósio Brasileiro de Informática na Educação. Curitiba: Sociedade Brasileira de Computação.

Azevedo, R., Martin, S. A., Taub, M., Mudrick, N. V., Millar, G. C., & Grafsgaard, J. F., 2016. *Are Pedagogical Agents' External Regulation Effective in Fostering Learning with Intelligent Tutoring Systems?* In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), Intelligent Tutoring Systems: 13th International Conference. Cham: Springer International.

Barcenes, A. B., Sanz, C., Baldassari, S., Ramos, E. R., Mendoza, G. V., Merino, R. B., Fierro, W, 2018. Efective computing: emotions, technologies and their relationship with virtual education. Revista de Investigación Talentos V.1.

Bernacki, M. L., Aleven, V., & Nokes-Malach, T. J., 2014. *Stability and change in adolescents' task-specific achievement goals and implications for learning mathematics with intelligent tutors*. Computers in Human behavior.

Curilem, S. G., Barbosa, A. R., de Azevedo, F. M., 2007. *Intelligent tutoring systems: Formalization as automata and interface design using neural networks, Computers & Education*.

Dazzi, R. L. S., 2007. *Metodologia para Adaptação de Interface e Estratégia Pedagógica em Sistemas Tutores Inteligentes*. Tese (Doutorado em Engenharia Biomédica) – Instituto de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis.

Dos Santos, C. T., Frozza, R., Dhamer, A., Gaspary, L. P., 2002. *DÓRIS — Pedagogical Agent in Intelligent Tutoring Systems*. In: Cerri S.A., Gouardères G., Paraguaçu F. (eds) Intelligent Tutoring Systems. ITS 2002. Lecture Notes in Computer Science, vol 2363. Springer, Berlin, Heidelberg.

Dougiamas, M., Taylor, P. C., 2002. Interpretive analysis of an internet- based course constructed using a new courseware tool called Moodle. 2002. Available: <http://dougiamas.com/writing/herdsa2002/>

Giraffa, L. M. M., 1999. *Uma arquitetura de tutor utilizando estados mentais*. UFRGS.

Gonçalves, E. J. T., 2009. *Modelagem de arquiteturas internas de agentes de software utilizando a linguagem, MAS-ML*. Dissertação (Mestrado Acadêmico em Ciência da Computação) – Universidade Estadual do Ceará. Centro de Ciência e Tecnologia.

Grawemeyer, B., Mavrikis, M., Holmes, W., Gutiérrez-Santos, S., Wiedmann, M., & Rummel, N., 2017. *Affective learning: improving engagement and enhancing learning with affect-aware feedback*. User Modeling and User-Adapted Interaction.

Haendchen Filho, A., 2017. *Sistemas Multiagentes na Internet: Uma arquitetura orientada a serviços para o desenvolvimento de agentes inteligentes de software na Internet*. Novas Edições Acadêmicas.

Kimball R., Merz R., 2000. *The Data Webhouse*

Haendchen, F. A., Prado, H. A. and Lucena, C. J. P., 2007. A WSA-based architecture for building multiagent systems. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS '07). ACM, New York, NY, USA.

Kimball, R., Merz, R., 2000. *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*. John Wiley & Sons, Inc.

Latham, A., Crockett, K., & McLean, D., 2014. *An adaptation algorithm for an intelligent natural language tutoring system*. Computers & Education.

Lin, H. C. K., Wu, C. H., Hsueh, Y. P., 2014. *The influence of using affective tutoring system in accounting remedial instruction on learning performance and usability*. Computers in Human Behavior.

Mayer, R. E., 2014. *The Cambridge Handbook of Multimedia Learning: Second Edition*. 2 ed. Cambridge University Press.

Positivo, U., 2017. *Manual do Aluno: Graduação EAD*.

Raabe, A. L. A., 2005. *Uma proposta de arquitetura de sistema tutor inteligente baseada na teoria das experiências de aprendizagem mediadas*. Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Informática na Educação.

Rissoli, V. R. V., Giraffa, L. M. M., Martins, J. P., 2006. *Sistema Tutor Inteligente baseado na Teoria da Aprendizagem Significativa com acompanhamento Fuzzy*. Informática na educação: teoria & prática, v. 9, n. 2.

San Pedro, M. O. Z., Baker, R. S. J. D., Rodrigo, M. M. T., 2014. *Carelessness and Affect in an Intelligent Tutoring System for Mathematics*. International Journal of Artificial Intelligence in Education, v. 24, n. 2.

Souza, C. V., Shiguti, W. A., Rissoli, V. R. V., 2013. *Metodologia Ativa para Aprendizagem Significativa com Apoio de Tecnologias Inteligentes*. Conferência Internacional sobre Informática na Educação.

Tomelin, J. F., 2016. *Perfil das Instituições que ofertam EAD no Brasil*. ABED.

VanLehn, K., 2006. *The Behavior of Tutoring Systems*. International Journal of Artificial Intelligence in Education.

Vicari, R. M., Giraffa, L. M. M., 2003. *Fundamentos dos sistemas tutores inteligentes*. In: BARONE, D. (org.). Sociedades Artificiais: a nova fronteira da inteligência das máquinas. Porto Alegre: Bookman.

Vos, H. J., 1995. *Applications of Bayesian decision theory to intelligent tutoring systems*. Computers in Human Behavior.

Wenger, E., 1987. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman.

Woolf, B. P., 2010. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.