Project number: 543301-LLP-1-2013-1-UK-KA3-KA3MP

ENACT KA3 – Multilateral Project

Enhancing Negotiation skills through on-line Assessment of Competencies and interactive mobile Training

Lifelong Learning Programme 2013
Leonardo da Vinci

D6.1. Report on artificial cognitive modelling and intelligent tutor: a Literature Review

Project start date: 01/01/2014 Duration: 24 months

Work Package 5 - Artificial intelligent agents and intelligent tutor
Leading Partner: Plymouth University

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Document Sign-off

Nature	Name	Role	Partner	Date
DRAFT	Daniela Pacella		PLYM	02.12.2014
REVIEWED	Elena Dell'Aquila		PLYM	03.12.2014
REVIEWED	Özgür Yaşar Akyar	QM	GSB	10/12/2014
APPROVED	Davide Marocco	PC	PLYM	14/12/2014

1. Introduction

In the recent years, the need of technologic platforms to assist traditional learning methodologies arose, and several experiments showed the importance of one-on-one instructions compared to one-to-many (Ong et al., 2003). Many of the first developed learning platforms based on curriculum sequencing do not provide the user a personalized experience but just rely on the user's knowledge and progress on the subject matter; the progress is mostly evaluated through direct questions. More recently, these kinds of systems were replaced by more adaptive ones, based on machine learning and AI techniques; these platforms, referred to as Intelligent Tutoring Systems, are able to give a more personalized curriculum and to assess the learner's knowledge on the matter; but several obstacles still prevent the wide use of these tools, among which the difficulty of finding the right way to deploy the information to the user and to collect data from the learner, the lack of evaluation standards to assess the user's knowledge and the time and cost of the software development. Also most of recently developed platforms do not take into account the user's cognitive and psychological characteristics, such as learning style, communication style and other related skills which could help the system to build a personalized environment and strongly improve the user's learning experience. In literature, these characteristics are referred to as "soft skills" and their importance is growing in different fields as business and SME market.

This review will explore the actual state of art of Intelligent Tutoring Systems, analysing strengths and weaknesses of the actual platforms and the barriers which prevent the widespread of these learning methods; it will offer a view on the different implementation methods, including situated tutors, game-based learning and serious games; it will also show the reasons behind the birth of ENACT Project and the actual urgent need to develop a methodology to assess the user's soft skills using a reliable and scientific method.

The potential of the platform developed by ENACT will be dependent on four primary objectives:

1) The development of a 3D game-based platform to assess the user's negotiation and communication skills in realistic scenarios during the interaction with an Intelligent bot;
2) The development of an Intelligent Tutor able to provide a personalized training on the user's negotiation skills;
3) The implementation of a system to provide a reliable and scientific-based assessment with the use of modern psychological models.
4) The development of a software completely free-to-use, compatible with all modern devices (PC, smartphones, tablets), handy to access for wide and heterogeneous groups of users.

2. Why an Adaptive Tutoring System?

The link between education and technology lays its eggs in the 50's, when the first teaching machines, with linear, unidirectional paths, were designed by the behaviourists. These systems were not able to build and sequence material or to provide a tailored feedback, and the way they were designed did not provide any specification on the type of instruction they were deploying but just on how to deploy it. This means that there was no difference in the architecture according to the kind of taught subject. These systems were not registering neither responding to the user answer to their instruction, since the path was predefined and there was no possibility to switch it or go on a previous step. Only in the late 60's some more complex systems started to emerge, which included non-linear paths and a slightly more tailored material structure, resulting, for instance, in evolved finite state machines with branch architecture. However, the implementation of these systems, despite their simplicity, had two main aims: a better understanding of the processes that lie behind education and a demonstration of the effectiveness of one-on-one instructions, even if deployed by a non-human system. In fact, also in recent years many studies demonstrated how effective one-on-one tutoring is, proving that students who receive this kind of instruction perform two standard deviations better than students in traditional classrooms (Ong and Ramachandran, 2003). But what kept this type of systems far from perfection was the lack of consideration of cognitive aspects, such as emotions, motivation and specific user temper. Beyond the idea of "one size fits all" there is a variety of shades into the user behaviour that cannot be ignored as the performance of the learner will strongly rely on these variables. The thought that not every learner could approach a subject in the same manner as the others lead to structure the teaching material considering the single student's needs and characteristics. Sleeman and Brown, in 1982 defined Intelligent Tutoring Systems (ITS) as "adaptive systems which use intelligent technologies to personalize learning according to individual characteristics such as knowledge of the subject, mood and emotion". A recent definition of Conati states that "ITS is the interdisciplinary field that investigates how to devise educational systems that provide instruction tailored to the needs of individual learners, as many good teachers do." Intelligent Tutoring Systems learn about the user while the user learns about the subject and their evolution is synchronous with his progress and the interaction with him. The field of ITSs is wider than it can seem and embraces a big range of intelligent systems, like conversational tutors, simulation-based tutors and modern serious games. As Conati underlines, "There are, however, other educational activities that can benefit from individualized computer-based support, such as studying examples, exploring interactive simulations and playing educational games. Providing individualized support for these activities poses unique challenges, skills and mental states often not as structured and well-defined as those involved in traditional problem solving." (Intelligent tutoring system: New challenges and directions, 2009) Many papers underline how the use of ITSs provides better results (Intelligent tutoring systems: past present and future, 1996) (Evaluation Methodologies for Intelligent Tutoring Systems). However, the higher the complexity of these systems is, the more time is required to develop them; it has been shown that in order to implement 1 hour of information deployed via an intelligent tutor, it is required an average of 200 hours of labour (Woolf, Cunningham, 1987). The cost of developing an ITS is incredibly high in terms of time and resources, especially considering the variety of disciplines which can be taught, the different purposes and user target. One of the biggest challenges of our time is, therefore, to build an Intelligent Tutoring System which can be used through different domains, from well-defined to ill-defined ones, that can cover strictly hard disciplines and soft skills, and that can be easily adapted to a wider range of users considering their sex, age and characteristics; also, its features should include accessibility, mobility and easiness to structure even for non-professionals.

3. The evolution of Tutoring Systems

When we speak about tutoring and assessment, we must not forget how integrated these two processes are. The concept of tutoring itself includes the ability to guide the learner and fill the "holes" in the learner's knowledge, and, in order to perform this, the tutor should be able to correctly diagnose the current state of the knowledge of the student, assessing in real time and modelling his own belief of the student. So an ITS is an attempt to reproduce in a computer what in a human would be called "good teaching" (Elsom-Cook, 1987). The ideal tutoring systems should be able to assess the student for his absolute knowledge of the subject, for his knowledge related to other learners (i.e. previous interacting agents) and for his progress during the session with the platform. To reach this point, the research on tutoring systems has gone far since its beginning. The first tutoring systems did not contain any representation of the learner and of the knowledge inside of it.

3.1. Computer-Assisted Instruction

The bases of Intelligent Tutoring Systems lie into Computer-Assisted Instruction, the first way to implement a tutoring system on a PC. These systems were structured to present the material to the learner on the screen in a programmed path. The first implementations had a linear structure, did not behave according to the user characteristics, and just "ignored" the user response, deploying the programmed information in the same way for every learner. They, in fact, do not possess a student module to build a representation of the student they are teaching to. This type of CAIs was called "ad-hoc frame oriented" (AFO) in order to specify its exclusive dependency on the author's implementation and material structure. Latter implementations of these systems were based on more Skinnerian principles of stimulus-response and for this reason they provided an immediate feedback about the correctness of the answers. The act of the tutor of switching his behaviour during the curriculum path and try to avoid another learner's mistake is called "remediation". In these simple systems, if the response provided by the student was incorrect, the CAI interrupted his curriculum path and presented the previous instruction again, followed by a stimulus similar by category and type to the former, in order to condition the student to the correct answer. The theory behind it was that, in disregard to the nature of the error, the student could have been conditioned to learn the right answer by the repetitive presentation of similar kind of stimuli, as stated by the behaviourist's "law of frequency". These programmed instructions systems dominated the 70's until the need of a more cognitive approach emerged.

3.2. Generative systems

Generative systems represent an evolution of CAIs, in a way that allows a more complex structuring of the teaching material from the system. This implies that the system is required to sequence material and to solve meaningful problems posed by the student. These systems, mostly used to deploy arithmetical and geometrical knowledge, represent the real precursors of ITSs: although they do not have a structured knowledge module, they are able to respond to the student's questions and to provide exercises tailored to the student's needs. The problem analysis and solution was generated by comparing parameters in the student's answer to a predefined list of possible outcomes, and then, according to the relevance, categorized the error providing the most useful exercise. It is considered the first type of system to provide a sort of scaffolding – conceptualised as "guided prompting that pushes the student a little further along the same line of thinking […] giving direct feedback on a student's response" (M. T. H. Chi et al., 2001, p. 490) – to the user through his learning process. The privileges of using this approach were the reduction of memory usage and a more dynamic management of the knowledge, even though the system was not "aware" of the subject matter. The instructions, in fact, were still stored as group of texts, without any categorization or modelling. Another name for this kind of systems was "adaptive systems", since they could adapt – in a simple but functional way - to the student's mistakes in the particular domain. The behaviour of these tutors cannot be considered "intelligent", since they lack of a student model, and also they are not able to provide dynamic solutions to problems. The next step would have been to add Artificial Intelligent techniques into Tutoring.

3.3. Curriculum sequencing

(Brusilovsky (1992)) The main aim of Curriculum Sequencing lies behind the need of deploying information to the learner in an order that suits their need and takes into account their existing knowledge, their personality, learning style, individual traits and context of work. This allows the student to receive a personalised flow of instructions and type of material in a way that is best tailored to him. This kind of system was the first one to include a sort of simplified student model, which stored the learner's responses and was able to infer their knowledge at each step. Some of the early ITSs totally relied on this structure, while in recent works curriculum sequencing is still integrated into more complex and advanced systems. This technique, for instance, is still widely used in well-defined domains, since a reliable algorithm to define the student knowledge is enough to provide an adaptive curriculum to the learner.

4. Intelligent Tutoring Systems: structure and modules

A total redesign of Tutoring Systems was made with the integration of Expert Systems and Artificial Intelligence into non-human tutors. A definition by Polson and Richardson (1988), list three core features an Expert Tutoring System must have in order to be defined an Intelligent Tutoring System:

a) It must have an expertise on the subject, that is, to have a dynamical representation of the subject domain that allows autonomous problem solving;
b) It must be able to store information about the student expertise on the subject which includes the knowledge prior to the interaction, his progress and a meaningful assessment per each step;
c) It must adapt on the learner, using curriculum sequencing to structure the material in order to reduce the difference between the expert's and the learner's knowledge.

More recent works investigated the pros and cons of automated tutoring systems compared to human tutoring. In particular, the main advantages of using ITSs are (Latham et al., 2012):

I. Curriculum sequencing: as discussed above, the ability to decide an algorithm to sequence the information presented in a way that suits the learner's needs is an extremely powerful tool that allows customised experience and reliable student assessment;
II. Intelligent solution analysis: this technique allows dynamical problem decomposition and a better analysis of the mistakes of the learner, providing a detailed and customised feedback (Pahl & Kenny, 2009);
III. Problem solving support: according to the constructivist approach, ITSs allow a deeper understanding of the subject by helping the learner to autonomously reach the solution to the problem (Redondo, Bravo, Ortega, & Verdejo, 2007).

On the other hand, the disadvantages of ITSs are (Ramirez et al., 2013):

I. Expensive research: the development of every ITS requires a team which includes psychologists, computer scientists, educators and experts of the subject, plus an amount of time which may vary but still bigger in average than the preparation of a class lecture;
II. Lack of intrinsic motivation and human contact: most of the first ITSs did not give any real context in which the deployed knowledge could be used and did not provide any source of reward during the interaction. The contact with a human tutor, also, can be motivating itself for the learner thanks to the positive feedback given by words, facial expression and social context.
III. Lack of focus in learning content: rather than focusing in the learning content, many of the first ITS developed focused on the structure of the algorithm, thus the quality of the deployed content was poor, incomplete or limited and difficult to use in broader contexts.
IV. Lack of evaluation standards: since the specificity of the deployed content and since the lack of availability of other non-human platforms it is still very difficult to

compare the results obtained by automated instructions and those by traditional courses.

Many of these problems have been faced and approached with different proposed solutions; one of the most important is the intrinsic motivation, which allows the student to be tied to the pc until the interaction with the tutor becomes effective enough to provide a progress of his knowledge. This has been achieved in several ways, as, for instance, in the field of serious games, where contexts have been introduced in the form of game scenarios and where the presence of an extrinsic motivation often results in the creation of challenges which involve personal aspects of the learner. Also simulation-based tutors aim at recreating a virtual environment as close as possible to real life situations, thus arousing the same feelings and emotions.

A pioneering project by Carbonnel, SCHOLAR, represents one of the very first attempts to create an ITS, and helped to delineate the actual structure of a general Intelligent Tutoring System. According to this model, an ITS is composed of three core modules: the Learner's module, the Expert Module and the Instruction Module. These three components are interconnected and each can communicate directly with the other two. Another model of ITSs was structured in the same years by Dede (Dede, 1986) and included a fourth module, the User Interface, giving the Instructional module a different name, "Pedagogical module", which had the aim of building a customised path of knowledge for the student using the most efficient pedagogical strategies. The User Interface, instead, being the most external module of all, allows the collection of information about the user in order to modify the existent relations among the components.

The Expert Module. The Expert Module is the component which contains the full knowledge of the domain. This knowledge is generally provided by a team of human experts of the subject. Once the content is defined, the way it is organized, coded and fragmented during the development of the ITS can be chosen among several options:

I. Sorting the instructions using a mathematical order

II. Sorting the content using human-like way of reasoning

III. Collecting data from the target group.

The way the content is deployed is chosen by the Instruction Module which will access to the Expert Module database in order to provide the best material suited to the student.

The Learner Module. The Learner or User Module is in charge to interpret all the data obtained by the interaction of the user with the interface in order to assess their performance. The evaluation provided is called Learner's diagnosis, and represents the basis on which the Knowledge Module builds an adaptive instruction/curriculum. According to Polson (1988), the diagnosis is based on three dimensions:

- The bandwidth. The bandwidth represents the width of the information about the student that the system has access to. It may be at three different degrees: mental state, when all possible information are available; intermediate state, when almost all the information through the interactions is available; final state, when only the last student answer is available to be processed by the system.

- Type of knowledge. The knowledge type of the subject to be taught. It can be distinguished into three different types: flat procedural, when the knowledge is only procedural with no subgoals: hierarchical procedural, when it is procedural but also contains subgoals; declarative.
- Student–Expert difference. This dimension concerns the gap between the student model and the expert module representation. The degree of complexity the system can have is tri-dimensional:
  1. The system is able to recognize the learner’s missing knowledge;
  2. The system is able to categorize and infer the learner’s incorrect information;
  3. The system is able to map the student’s errors in a dynamical way.

The Instruction Module. The Instruction module, or the Pedagogical module in Dede’s notation, ensures that the information is deployed to the student in the more efficient way. The structured material, that will be the outcome of this module, must have this three attributes:
1) Flexibility: every instruction can be adapted to multiple purposes and logical order;
2) Structural transparency: every instruction must be self-explanative and structurally identical to the others;
3) Individualization: every instruction must be implemented in different ways that can be deployed to different kind of users.
The instructions should be implemented in a modular logic, dividing the material into units with small instructional goals and ensuring that in any case each unit achieves the goals.

The Environment. The Environment is the part of the system specifying or supporting the activities that the user does and the methods available to the user to do those activities. Introduced in the fourth modules model (Dede, 1986), it includes both the User Interface and the more general setting in which the user is supposed to access the platform.

5. Designing an Intelligent Tutoring System

5.1. Well-defined vs ill-defined domains

(Leveraging a Generalized Tutoring Framework in Exploratory Simulations 62 Of Ill-Defined Domains James Thomas, Ajay Divakaran, and Saad Khan) While nearly all of the former literature on Intelligent Tutoring Systems regarded building systems for well-defined domains, where there was a black-and-white distinction between what was wrong and what was right, one of the main challenges of designing tutoring systems regarded the need to structure platforms able to face the problem of the multiplicity of possible student behaviour that have shapes of correctness which may vary according to the concept, the contest and other involved agents. Intelligent Tutoring Systems for ill-defined domains differ on the former systems mainly on the following points:

1. It is needed a certain amount of information regarding the student in order to retrieve the correct diagnosis and assessment;
2. The system's feedback on the correctness of the answer is delayed until a partially complete student model has been made.

Most of the tasks posed to the student do not have a defined answer and this ambiguity is reflected in the complexity of the interaction between the tutor and the student, the need of shaping the student model through time and the constant endeavour through the interaction to catch the granularity of the aspects which determine the behaviour of the user.

5.2. Diagnosis and adaptation

As we mentioned earlier, the diagnosis of the student knowledge must take into account several dimensions and information which must be gathered throughout the interaction of the user with the interface. In order to process and categorise these inputs, there are some commonly used heuristics which allow an assessment and a prediction of the student knowledge.

5.2.1 Rules-based models

Rule-based ITSs are based on the idea of establishing a set of rules that a student should follow while finding a solution to the problem presented. Properties of rule-based systems, according to Hayes-Roth are (Hayes-Roth, 1985):

1. The ability to incorporate practical human knowledge in conditional IF-THEN rules;
2. Their skill increases at a rate proportional to the enlargement of their knowledge bases;
3. They can solve a wide range of possibly complex problems by selecting relevant rules and then combining the results in appropriate ways;
4. They adaptively determine the best sequence of rules to execute;
5. They explain their conclusions by retracing their actual lines of reasoning and translating the logic of each rule employed into natural language.

Every learner error should be predicted according to this model, and the system can provide a feedback which helps the user to identify the right steps to follow.

5.2.2 Case-based models

In case-based models, the student's errors are not coded into IF-THEN rules but in more meaningful cases, which are more specific, related and reveal a strategy. The cases are built gaining data from a large amount of learners with the help of machine learning techniques.

5.2.3 Constraints-based models

The concept of constraint-based systems lies in Ohlsson theory of the difference between declarative and procedural knowledge, in which the former indicates the pure knowledge of the subject while procedural knowledge allows the generation of actions. The declarative knowledge is important to evaluate the consequence of the actions, and therefore to expect a certain outcome. According to this model, the only way for the user to learn from mistakes is to create constraints on the possible correct answers, generating set of rules that give a feedback in order to restrict the range of the declarative knowledge of the user.

The main difference between rule-based systems and constraint-based systems lie in the fact that the latter concentrates on the obtained result in each of the steps to reach the solution to the problem and does not require a global Expert Module to process the sum of the information required. Assuming that the constraints are respected, every solution that solves the problem correctly can be considered satisfying, and, for this reason, each step can be auto-conclusive, denying the scope of a wider goal. This is the main reason why Constraint-based models are usually applied in ill-defined domains.

6. Expert Systems

Regarding the integration of Expert systems into ITS, there are many examples in literature of researches which used AI techniques as Bayesian networks, Neural networks and Fuzzy sets to allow the systems to be autonomous, self-adaptive and “intelligent” in the concept as we know it.

6.1. Bayesian Networks

Costa and colleagues used Bayesian Networks to design a system which possesses an Open Learner Model, in the sense that it allows the student to be involved in the creation of his own Module (costa et al., 2012). This approach leads to a more dynamical interaction between the Instruction Module and the user, in order to improve the efficiency and the time needed for a reliable assessment. The system uses two Bayesian networks which are structurally identical, one to compute the learner's belief in his knowledge (Ms) and one to compute the tutor's belief in the learner's knowledge (Mt). Each of these has two parts, the domain-general circuit, which computes the general belief throughout the whole interaction, and the task-specific circuit, which is created at run-time and concerns the problem faced by the student at the moment. When the student has finished solving a task, the probability and data obtained are directed as input to the domain-general circuit, modifying the structures of the nodes. Every node of the networks have two values: mastered or not-mastered; the weights are established within 5 possible degrees of knowledge: No idea=0.05, Basic=0.25, Good=0.50, Very good=0.75 and Expert=0.95. At the end of each task, instead, the learner is asked is belief regarding his degree of knowledge which may have the following values: Very unsure=0.05, Unsure=0.25, Almost sure=0.5, Sure=0.75 and Very Sure=0.95. If the beliefs contained in Ms and Mt differ, the tutor starts to negotiate with the student in order to find an agreement. The negotiation process can include tasks like proofing, by solving or analysing problems posed by the tutor, that the tutor's belief is wrong. If the learner believes he has less knowledge than estimated by the system, a support strategy will be engaged in order to convince the learner that he is more proficient than he thinks.

6.2. Neural Networks

Only few authors in literature used neural networks to predict student models. This mainly happens because, even if neural network are extremely good at both categorising unpredictable situations and learning from data, it is not easy at all to retrieve bug libraries in order to train them. This situation is slowly changing especially since the growing interest in big data science, but still it is needed a huge amount of work in order to give the right inputs and obtain generalizable data out of artificial neural networks.

One of the first approaches used a C programmed feedforward neural network with three layers to compute subtraction problems (Mengel et al., 1992). The neural network was trained using an already available set of bugs.
A more recent approach by Wang and Mitrovic describes the use of data collected by SQL-Tutor, a constraint-based model tutoring system, to train a neural network in order to map student errors (Wang et al., 2002). The neural network was feedforward and its input layer took the following parameters:
- the time needed by the student to reach the solution;
- the level of help requested by the student;
- the level of complexity of the posed problem;
- the current level estimated of the student.

After the training session, the network was used to predict the behaviour of the students. It was found out that six submissions of the student answers were requested in order for the network to assess his behaviour properly and to select an appropriate problem afterwards.

Lo and colleagues recently used a multilayer feedforward neural network to identify the users' cognitive styles in a web-based learning environment (Lo et al., 2012). This methodology represents another trial to gather information about the learners' psychological traits in an unobtrusive way, without the use of psychometrical explicit tests and thus without separating the assessment from the learning process itself. The designed system is able to analyse the user browsing behaviour, to infer a cognitive style out of it and to sort the educational path according to it. Since the importance for the authors to design a system that collects information about the users in a totally implicit way, there is no need for a login and only the material for each session is stored, i.e. when the user disconnects and then logs back in, it is not possible to recover the previous data. The model of cognitive styles adopted in this work is the one by Myers-Briggs, which is based on Jung's theory of cognitive types and has four possible profiles: Interpersonal, Mastery, Understanding and Self-expressive. In the first part of the project, they collected data to investigate the relationship between the browsing behaviour and the cognitive styles and then used these data to train the neural network. The neural network was three-layered and its input neurons received data from:

- Ratio and frequency of the selection of contents and interactive components;
- Average staying time on content and interactive components;
- Selection ratio of content link types;
- A bias unit.

The output units were four and the activation of each of them returned a different profile. The approach used by the authors is extremely interesting, though their results cannot be generalised for several reasons: the students used for the experiments were not in the same classes and had different teachers; there is no demonstration of an increased attention paid to the contents when the system adapted to the identified style, but only of a very low increasing of the staying time on the webpage. Also, identifying the cognitive style did not take into account the motivations of the single user.

6.3. Fuzzy sets

Fuzzy sets are incredibly good for adding a “human-like” way of reasoning to computers. Among the main conceptualizations of the fuzzy logic, Zadeh lists some of the features and properties this logic has (Zadeh et al., 1992):

- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
- In fuzzy logic, everything is a matter of degree.
- Any logical system can be fuzzified.
- In fuzzy logic, knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
- Inference is viewed as a process of propagation of elastic constraints.

The use in literature of fuzzy inference systems is wide, especially concerning ITSs.

A recent work by Hsieh and colleagues aimed at building a system to support foreign languages learning for Taiwanese people (Hsieh et al. 2012). The system aims at suggesting tailored English article by using the data gathered through multiple interactions with the user. Fuzzy sets and memory cycle updates are used in order to build the learner's preference. A tutoring system is implemented through an analytic hierarchy process to provide a motivating environment for English learners. Both the ability of the user and their vocabulary cognition are evaluated. Right after reading a new article, the user is asked to answer to a questionnaire about their motivation, satisfaction and feedback. The learner feedback about the understanding of the suggested article was collected using a five point Likert scales with values that indicated “Not at All”, “Not All”, “Moderately”, “Well”, and “Very Well”. Also a test asks about the newly learned words in order to help the memory to retain the information. The next suggested article will take into accounts both new words and previously learned words, to efficiently improve the volume of the vocabulary of the learner. To calculate the volume of the vocabulary of a new user, they are asked to answer a set of 10 questions using the Computerized Adaptive Testing (CAT) approach.

A modern approach to intelligent teaching in software pattern design is DEPTHS (Jeremic et al., 2012), a platform to build and educational path for each student taking into account their ability, background and characteristics. Most of the actual platforms use stereotype-based approaches in order to initialise the tutor in his first approach with a new user. This means that the first interactions with the student will always follow a predefined pattern and, once the system has collected enough data to build a student model out of the user, then the process of modifying the existing stereotype starts. This may lead to problems as lack of flexibility and lack of creating new stereotype by learning from the interaction with users. The approach used by the authors, instead is the overlay-based model. According to this approach, the student model is a subset of the general which can be just a part of it (in the strict overlay approaches) or a misconception of it (in the extended overlay approach). In DEPTHS, each domain concept is associated with the student's knowledge status in relation to that concept. The computation of this state is based on fuzzy sets and certainty factor theories. Curriculum planning is left to Jess's inference engine, which computes the right concept to follow next according to four rules:

- student's current knowledge state should be the same as or lower (if possible) than the minimal knowledge level predefined;
- for learning that concept;
- the concept should not be learned earlier;
- all prerequisite concepts should be already learned;
- the concept should not be already in the concept plan.

A Diagnostic engine uses a set of pedagogical rules and domain knowledge in order to diagnose test results and infer the student's knowledge level based on these results.

7. Conversational Intelligent Tutoring Systems

Conversational Intelligent Tutoring Systems (CITSs) are platform which aim at replicating the interaction of students with real teachers, as in classrooms, with the introduction of natural language. The development of such platforms is extremely time-consuming and the speech recognition must be at a high level in order to prevent the student to abandon the platform for lack of motivation. For these reasons, there are only few platforms which use conversational agents to communicate with the user.

An example of this kind of platforms is OSCAR (Latham 2012). Oscar Conversational Intelligent Tutoring System is an ITS developed in order to implicitly predict and adapt to the user's learning styles using natural language recognition The learning style characteristics are taken from the Index of Learning Styles (ILS) model by Felder and

Silverman, according to which the learning style of a subject can be described by the score obtained in a dedicated 44-items questionnaire along four dimensions: perception (sensory vs. intuitive), input (visual vs. auditory), processing (active vs. reflective) and understanding (global vs. sequential). The model also describes the typical behaviour expressed by each prevailing style. The aim of OSCAR CITS is not to just predict and assess the user learning style, (as already done in Cha et al., 2006), but to imitate the behaviour of a human tutor during a conversation and presenting the material tailored to the learner, giving adaptive hints and analysing the problems posed. The CITS architecture detected and categorised behavioural cues using a set of 33 logic rules to apply to the information gathered. The first implementation was used to teach an online course of SQL and .net framework.

A more complex and complete system is INES (Mikic Fonte et al., 2012). The INtelligent Educational System (INES) is a prototype of an e-learning platform which has the aim to deploy information, manage multiple users, assess the learners, guide the learner and grasp natural language speech using multiple virtual agents. In order to do that, the platform uses the following set of tools:

- BDI technology agents, according to the BDI model. The BDI (Beliefs, Desires, and Intentions) model is based on the idea of logical agents which possess the knowledge of the actual state (the Beliefs) and has a set of goals to achieve (Desires). If the knowledge of the current situation differs from the one expected by the set goals, the agent has a list of plans to execute in order to try to reach the desired state. Multiple and different agents compose the ITS of the INES platform;
- Jadex platform, a tool which allows easy programming and initialization of virtual agents following specific features and variables set by external coders. The agents created by Jadex are known as Goal Oriented Agents, and the possible actions they can execute are written in Java and XML;
- CHARLIE, a chatterbot which allows natural speech recognition.

In particular, the structure of INES provides that, for every new user connected to the platform a new instance of a BDI agent inside the Expert Module of ITS is created, which dynamically starts adapting to the user and stores the information gained. The user, throughout the interface, does not communicate directly with the system. First of all the natural language processing goes through a first agent, a chatterbot, which analyses the words and the concepts used by the user. This information is then delivered to a particular BDI agent called EMMA, which elaborates the information and decides whether to send those to the Expert Module for a decision-making process or send a response back directly to the user without a higher level processing. If a higher order message (as "the user has completed the session") must be sent to the user, it overrides the messages sent by the other two agents. If the information gained by EMMA needs a different level of processing they are sent to the BDI agent correspondent to the user inside the Expert Module, which is called ISMAEL (Intelligent System Manager Agent for E-Learning). The instance of ISMAEL for each user contains their educational progress, their credentials and personal information. The whole platform organization and information exchange is able to plan the educational path for each user and suggest them personalized tasks to achieve specific learning goals. What lacks in these kinds of systems, as already evidenced by the authors, is the presence of an effective intelligence showed by the agents; other than that, the system seems not to take into account emotional aspects of the user interacting with these agents.

8. Situated tutors

The Interactive Strategy Training for Active Reading and Thinking-Motivationally Enhanced (iSTART–ME) tutor is a recently developed game-based platform to enhance self-explanation of science text in young adolescents (Jackson et al., 2012). This serious game is built on top of an existing ITS which had the same aim, but that required too long interactions in order to have long-lasting results, thus reducing the motivation to stay. There are several modules in this platform:

- the Introduction Module, during which three virtual agents introduce the concept of self-explanation by making examples and providing strategies for an efficient reading;
- the Demonstration Module, where two characters demonstrate self-explanation by reading science texts and the student must guess the strategies used in these examples. A tutoring character gives a feedback about the self-explanation given by the other virtual agent.
- the Practice Module, where the student is asked to use the acquired knowledge to use self-explanation to understand the proposed science texts.

Another example of the gamification of an ITS for children is described in the work by Sandberg and colleagues (Sandberg et al., 2014). In the work they presented they developed a motivationally enhanced version of the already existing MEL platform, an application to improve the native English speaker's vocabulary of the animal and continents words. There were 25 stimuli of animals and 5 continents, along with embedded videos and short stories about the animals. Several kinds of tests were presented to the children, among which multiple choice quizzes, yes or no, puzzles and spelling questions. The enhanced version of the application contained two different game modes:

- Zoo Animals: there is a storyline in which the animals escape and need to be found back.
- Neighbourhood: items need to be found in a collage.

The authors tested three hypotheses: 1) Children who used MEL-enhanced outperformed those who used the simple version of MEL; 2) Children who used MEL-enhanced learned in a more efficient way than children who use MEL-original; 3) Children who use the MEL-enhanced spent significantly more learning time on the platform. While the hypotheses 1 and 2 have been verified, the third revealed to be false.

Another kind of modern platform which exploits most of the tools previously mentioned is the architecture described in Giuffra et al., 2013. This model wishes to integrate a Virtual

Learning Environment and ITSs techniques to create responsive agents which can guide the learner throughout their interaction with the teaching material. The tool used for the Expert Module form is Moodle, a well-known and common platform to build MOOCs, interactive teaching environments and it is fully customisable, allowing researchers and teachers to save log files of the users. This platform is massively used by institutional websites. The Pedagogical Module is held by a virtual agent called “Bedel”, which has access to the database in order to retrieve relevant information to get the material sorted. The Instruction (tutoring) module is set into the form of an agent called “Tutor”, which takes the information both from the database and from Bedel, allowing it to provide suited and efficient educational path to the learner. Both the agents are coded using the already mentioned BDI model, using the Jason platform, a program in Java.

The ENACT Project

In the actual context, where most of the platforms are extremely well designed, implemented, and rich in its potentials, like in the case of DOKEOS Game (www.dokeos.com), a game environment which allows fully customised scenarios where to practice interactive skills and communication styles, what really emerges is the lack of standardised technologic tools to assess (neuro)psychological characteristics and deficits. Paper and pencil tests are still widely used even by human resources managers, though these kinds of tests require time to be completed, time to be interpreted, money, double processing, the opinion of experts and they result in a slow candidate selection and assessment. The European Project ENACT (Enhancing Negotiation skills through on-line Assessment of Competencies and interactive mobile Training) aims at developing a 3D environment where users can receive scientific, standardised and reliable profiles of their negotiation skills in a quick and handy way.

The ENACT platform will be composed of two main features:

1. A virtual agent which will be able to interact effectively with the user, answer according to their responses and have a dynamic behaviour;
2. An intelligent tutoring system, which will evaluate the interaction and the performance of the user within the scenario and will give a customized profile; other than that, it will allow the user to navigate through the history of the interaction with, in order to provide a molar and a molecular view on the choices taken by the two agents.

The virtual 3D agent will be implemented in a game-based scenario developed within the Unity platform. It will be able to express a range of basic emotions in an effective way using verbal cues, as vocal tone and structure of the sentence, and non-verbal indicators, as facial expression, eye contact, body posture and gestures. The interaction with the user is organized into states, which include one turn of speech for each party. The non-verbal emotional expressions of the bot are coded with scripts of movements the agent can execute consistently to the meaning of the chosen sentence and their efficacy and unambiguousness will be standardized during the demo sessions also with the use of the

data we will collect through online and paper questionnaires administered to users at the end of every game session.
The bot possesses a series of internal states which vary during the progression of the interaction and influence its behaviour. Every artificial agent has a value for his “concern for self” and a value for his “concern for others” as provided by Rahim and Bonoma‘s bidimensional model of negotiation.
The tutoring system will be implemented to intervene at the end of the game session and provide useful information to the user about his behaviour related to the bot plus more static traits. The user will be given a profile out of the 5 as stated in Rahim’s model and tips about how to improve the efficacy of communication styles. The profile will emerge by comparing the behaviour of the user and the character of the bot he interacted with. The user will also be provided a history of all the choices made during the scenario and will be guided through the understanding of the possible hidden aspects of the negotiation both in an overview and state by state. The user will be invited to consider every aspect of the interaction, to take into account the bot’s reactions and to which extent these can influence the whole negotiation process.
The interface will be as intuitive as possible, and its usability will be measured. It will contain a basic and a more advanced customization panel for the user to choose the non-verbal expressions in each state. The platform will be accessible both on mobile devices and on ENACT Game website and free to access.

9. Conclusions

In this review the actual state of art in literature of Intelligent Tutoring Systems has been explored, underlining which of the early systems’ features have evolved through time and which techniques are the most efficient and used among modern developers in this complex and changing field inside the Artificial Intelligence framework. The actual gaps between technology and theory have been highlighted in order to identify the possible directions of future research in the area; huge and fast progresses have been made since the first Intelligent Systems in the 80’s, but there is still plenty of dark corners in need to be explored, some of which will be shortly summarised below:

1. The lack of platforms which investigate in a reliable and scientific way the same category of variables currently assessed by paper and pencil psychological tests;
2. The lack of a standardised and agreed framework for the assessment of soft skills, which are typically distinguished by their undefined boundaries;
3. The lack of research on the emotions in virtual agents, both as they are perceived by the user and as the meaning they represent in comparison to the same categories expressed by non-virtual agents.

Other open challenges involve the ability to find models in which to code and frame ill-defined domains, which include the need of creative solutions in problem solving, facing unexpected situations, deal with overlapped contents and definitions, and learn from the experience with users. In this direction many developers are extending their user data analysis techniques from psychometrics to cloud computing and machine learning, allowing a deeper understanding of the user behaviour and a more meaningful use and access of many related variables whose real correlation is yet unknown.

10. References

Brusilovsky, P. (1992). A Framework for Intelligent Knowledge Sequencing and Task Sequencing.
In: *Proceedings of Intelligent Tutoring Systems*, 499–506.

Buckley, J. J., Hayashi, Y., & Czogała, E. (1999). On the equivalence of neural nets and fuzzy expert systems. *Fuzzy Sets and Systems*, 100, 145-150.

Cha, H. J., Kim, Y. S., Park, S. H., Yoon, T. B., Jung, Y. M., & Lee, J. H. (2006, January). Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In Intelligent tutoring systems (pp. 513-524). Springer Berlin Heidelberg.

Costa, E. d. B., Silva, P., maglhaes, J., Silva, M. (2012). An Open and Inspectable Learner Modeling with a Negotiation Mechanism to Solve Cognitive Conflicts in an Intelligent Tutoring System. in Herder, E., Yacef, K., Chen, L., and Weibelzahl, S., editors, *UMAP Workshops*, volume 872 of CEUR Workshop Proceedings.

Dede, C. (1986). A review and synthesis of recent research in intelligent computer-assisted instruction. *International Journal of Man-Machine Studies*, 24(4), 329-353.

Fancsali, S. E., Ritter, S., Stamper, J., & Nixon, T. (2013, July). Toward “Hyper-Personalized” Cognitive Tutors. In *AIED 2013 Workshops Proceedings* Volume 7 (p. 71).

Giuffra, P., Cecilia, E., Silveira, R.A. (2013). A Multi-agent system model to integrate virtual learning environments and intelligent tutoring systems. *International Journal of Artificial Intelligence and Interactive Multimedia* 2(1), 51–58.

Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, 28(9), 921-932.

Hsieh, T. C., Wang, T. I., Su, C. Y., & Lee, M. C. (2012). A Fuzzy Logic-based Personalized Learning System for Supporting Adaptive English Learning*. Educational Technology & Society*, 15(1), 273-288.

Huang, A. F., Wu, J. T., Yang, S. J., & Hwang, W. Y. (2012). The success of ePortfolio-based programming learning style diagnosis: Exploring the role of a heuristic fuzzy knowledge fusion. *Expert Systems with Applications*, 39(10), 8698-8706.

Jackson, G. T., & McNamara, D. S. (2013). Motivation and Performance in a Game-Based Intelligent Tutoring System. *Journal of Educational Psychology*, Vol 105(4), 1036-1049.

Jeremić, Z., Jovanović, J., & Gašević, D. (2012). Student modeling and assessment in intelligent tutoring of software patterns. *Expert Systems with Applications*, 39(1), 210-222.

Juárez-Ramírez, R., Navarro-Almanza, R., Gomez-Tagle, Y., Licea, G., Huertas, C., & Quinto, G. (2013). Orchestrating an Adaptive Intelligent Tutoring System: Towards Integrating the User Profile for Learning Improvement. *Procedia-Social and Behavioral Sciences*, 106, 1986-1999.

Latham, A., Crockett, K., & McLean, D. (2014). An adaptation algorithm for an intelligent natural language tutoring system. *Computers & Education*, 71, 97-110.

Latham, A., Crockett, K., McLean, D., & Edmonds, B. (2012). A conversational intelligent tutoring system to automatically predict learning styles. *Computers & Education*, 59(1), 95-109.

Lo, J. J., Chan, Y. C., & Yeh, S. W. (2012). Designing an adaptive web-based learning system based on students' cognitive styles identified online. *Computers & Education*, 58(1), 209-222.

Mengel, S., & Lively, W. (1992, March). Using a neural network to predict student responses. In Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's (pp. 669-676). ACM.

Mikic Fonte, F. A., Burguillo, J. C., & Nistal, M. L. (2012). An intelligent tutoring module controlled by BDI agents for an e-learning platform. *Expert Systems with Applications*, 39(8), 7546-7554.

Nkambou, R., & Bourdeau, J. (2010). *Advances in intelligent tutoring systems* (Vol. 308). Heidelberg: Springer.

Noh, N. M., Ahmad, A., Halim, S. A., & Ali, A. M. (2012). Intelligent Tutoring System using Rule-based And Case-based: A Comparison. *Procedia-Social and Behavioral Sciences*, 67, 454-463.

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4), 251-277.

Nye, B. D. (2013, July). Integrating GIFT and AutoTutor with Sharable Knowledge Objects (SKO). In *AIED 2013 Workshops Proceedings* Volume 7 (p. 54).

Olney, A. M., Hays, P., & Cade, W. L. (2013, July). XNAgent: Authoring Embodied Conversational Agents for Tutor-User Interfaces. In *AIED 2013 Workshops Proceedings* Volume 7 (p. 137).

Paviotti, G., Rossi, P. G., & Zarka, D. (2012). *Intelligent Tutoring Systems: an Overview*. Pensa Multimedia.

Polson, M. C., & Richardson, J. J. (Eds.). (2013). *Foundations of intelligent tutoring systems*. Psychology Press.

Popescu, E. (2010). Adaptation provisioning with respect to learning styles in a Web-based educational system: an experimental study. *Journal of Computer Assisted Learning*, 26(4), 243-257.

Posey, C. L., & Hawkes, L. W. (1996). Neural networks applied to knowledge acquisition in the student model. *Information sciences*, 88(1), 275-298.

Ray, C., & Gilbert, S. (2013, July). Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine. In *AIED 2013 Workshops Proceedings* Volume 7 (p. 37).

Rowe, J, Lobene, E., Sabourin, J., Mott, B., Lester, J.(2013). Run-Time Affect Modeling in a Serious Game with the Generalized Intelligent Framework for Tutoring. In *AIED 2013 Workshops Proceedings* Volume 7 (p. 95).

Sánchez-Vera, M. D. M., Fernández-Breis, J. T., Castellanos-Nieves, D., Frutos-Morales, F., & Prendes-Espinosa, M. P. (2012). Semantic Web technologies for generating feedback in online assessment environments. *Knowledge-Based Systems*, 33, 152-165.

Sandberg, J., Maris, M., & Hoogendoorn, P. (2014). The added value of a gaming context and intelligent adaptation for a mobile learning application for vocabulary learning. *Computers & Education*, 76, 119-130.

Santhi, R., Priya, B., & Nandhini, J. M. (2013). Review of intelligent tutoring systems using bayesian approach. *arXiv preprint* arXiv:1302.7081.

Schatz, S., Oakes, C., Folsom-Kovarik, J. T., & Dolletski-Lazar, R. (2012). ITS+ SBT: A review of operational situated tutors. *Military Psychology*, 24(2), 166-193.

Shute, V. J., & Psotka, J. (1994). Intelligent Tutoring Systems: Past, Present, and Future (No. AL/HR-TP-1994-0005). ARMSTRONG LAB BROOKS AFB TX HUMAN RESOURCES DIRECTORATE.

Sina, S., Kraus, S., & Rosenfeld, A. (2014). Using the Crowd to Generate Content for Scenario-Based Serious-Games. *arXiv preprint* arXiv:1402.5034.

Stathacopoulou, R., Magoulas, G. D., & Grigoriadou, M. (1999). Neural network-based fuzzy modeling of the student in intelligent tutoring systems. In *Neural Networks*, 1999. IJCNN'99. International Joint Conference on (Vol. 5, pp. 3517-3521). IEEE.

Stern, M. K., & Woolf, B. P. (1998, January). Curriculum sequencing in a web-based tutor. In *Intelligent Tutoring Systems* (pp. 574-583). Springer Berlin Heidelberg.

Thomas, J. M., Divakaran, A., & Khan, S. (2013, July). Leveraging a Generalized Tutoring Framework in Exploratory Simulations of Ill-Defined Domains. In *AIED 2013 Workshops Proceedings* Volume 7 (p. 62).

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197-221.

Vos, H. J. (1995). Applications of Bayesian decision theory to intelligent tutoring systems. *Computers in human behavior*, 11(1), 149-162.

Wang, T., & Mitrovic, A. (2002, December). Using neural networks to predict student's performance. In Computers in Education, 2002. Proceedings. International Conference on (pp. 969-973). IEEE.

Yager, R. R., & Zadeh, L. A. (Eds.). (1992). An introduction to fuzzy logic applications in intelligent systems (pp. 69-93). Boston: Kluwer Academic.