ITSEGO: An Ontology for Game-based Intelligent Tutoring Systems

Valentina Centola[1] and Francesco Orciuoli[2]
[1]*Istituto Comprensivo Gonzaga, Via Caduti di Bruxelles, Eboli (SA), Italy*
[2]*DISA-MIS, University of Salerno, Via Giovanni Paolo II, Fisciano (SA), Italy*

Keywords: Intelligent Tutoring Systems, Game-based Learning, Storytelling, Ontologies.

Abstract: This work proposes the definition of a tool supporting the transition of children from kindergarten to primary school and, as a side effect, the development of problem solving and digital competences. The tool has been defined, by means of an ontology-driven approach, as an Intelligent Tutoring System (ITS) integrated to a structured game-based educational environment and provides benefits for both teachers and children. The definition of a novel ontology, namely ITSEGO, providing a model (generally applicable in different learning contexts) to build Game-based ITS and, the design of a concrete Game-based ITS for supporting the aforementioned transition are the main results of this work.

1 INTRODUCTION

In the last twenty years, numerous research activities emphasize that the human learning process is not sequential but rather it exploits a number of conceptual interconnections (Collins and Loftus, 1975) forming a network. In this scenario, Information and Communication Technologies (ICT) provides a significant contribution in supporting (e.g., visualizing, organizing, personalizing, adapting and sharing learning content), improving (e.g., increasing engagement of students) and making possible (e.g., satisfying special needs) learning and teaching activities with respect to different subject matters. One of the most important usage of ICT for supporting learning and teaching activities is represented, without any doubt, by Intelligent Tutoring Systems (ITS). ITS (Polson and Richardson, 2013) are software systems providing adaptive educational experiences for both students and teachers. ITS have a great potential to improve school education, at the moment, unexpressed yet. Adaptivity is supported by ITS in several ways: i) providing students' learning activities coherently with their current knowledge and skills in order to foster meaningful learning; ii) providing individualized feedback able to stimulate next learning activities and avoid frustration, demotivation and disengagement due to unsuccessful performances; iii) providing hints helping students during the execution of their learning tasks. Adaptation capabilities of ITS and the evidence related to the improvement of learning experiences, when game-based approaches are used (Jackson and McNamara, 2013), lead to the idea to integrate these two elements in order to synergistically exploit the advantages of both. In several works like (McNamara et al., 2010) ITS are enhanced with game-based features in order to increase engagement of learners. Otherwise, this paper proposes to inject ITS features into game-based environments for children. Thus, we introduce a novel ontology for modeling such features. Such ontology can be populated in order to develop a concrete ITS-empowered educational game. Although ontologies have been studied extensively with respect to ITS (Mizoguchi et al., 1995), at the moment there are not ontologies, based on Semantic Web technologies, modeling all the aspects related to an ITS and able to achieve different objectives like those reported in Section 3.2. It is important to underline that the main part of this work can be generally applicable for a wide range of targets (also, for instance, to training) and heterogeneous domains.

2 OVERALL APPROACH

The main idea of the paper is to provide a tool for supporting children in the transition from kindergarten to primary school. In our understanding, technological environments that blend games, stories and intelligent (automatic) tutoring, represent effective solutions for the above mentioned aim. In this section we will motivate this idea and provide the necessary background knowledge. In the next sections we will

Centola, V. and Orciuoli, F.
ITSEGO: An Ontology for Game-based Intelligent Tutoring Systems.
In *Proceedings of the 8th International Conference on Computer Supported Education (CSEDU 2016) - Volume 1*, pages 238-245
ISBN: 978-989-758-179-3
Copyright © 2016 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

define a specific Game-based ITS, by which it is possible to achieve our goal, through two main phases. The first phase is the definition of ITSEGO (Intelligent Tutoring System for Educational Games Ontology), an ontology for Game-based ITS. This ontology can be generally applicable to a wide range of learning contexts. The second phase is the adoption of ITSEGO (its instantiation and the provisioning of additional contents) to produce such specific Game-based ITS. Thus, the second phase proposes an ontology-driven process to build Game-based ITS.

2.1 Motivating ITS and Game-based Learning Environment

This work is focused on children during their last year of kindergarten. They face a critical period with respect to their personal development and they are going to face the first class of primary school. In the context of such class they will develop also competences needed for future learning. Despite of the importance of the above mentioned issues, there are no concrete recommendations or guidelines, coming from the Italian Ministry of Education, which support teachers in evaluating the development of these competences. On the other hand, since 2004, the required age to be admitted to attend the first class of primary school has been decreased. In fact, children born until 30 April (of the school year) can begin such school. This kind of procedure is executed, on demand of the parents, after discussing with the teachers of their children.

On the basis of these considerations, this work starts from the idea of ITaG (McNamara et al., 2010) and proposes a concrete framework to adopt systems, which integrate intelligent tutoring mechanisms and game-based learning environments, for supporting the above mentioned transition.

The main benefits of using such systems are: i) ITS allow children to develop basic competences concerning their personal growth in a personalized and incremental way, meanwhile, the game-based component of the whole system allows to increase engagement and motivation, makes smooth the transition between the two considered types of school and, lastly, to implement the learning-by-doing approach; ii) ITS could trace all the child's actions and provide structured data to be analyzed by teachers in order to execute better evaluations; iii) ITS and (digital) game-based learning environments help children to improve digital competences, which are included in the key competences recommended by the European Parliament in 2006[1].

[1]http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=URISERV:c11090

2.2 Fundamentals and Background Knowledge

This section provides the reader with key concepts of ITS and Game-based Learning Environments. These two concepts will be the foundation for this work.

2.2.1 Intelligent Tutoring Systems

An ITS (Polson and Richardson, 2013) is a software agent whose behavior can be, informally, described by an outer (external) loop and an inner loop (du Boulay, 2006). The external loop proposes to the learners a sequence of *tasks* (problem solving) of different difficulties. The default behavior is that the next task has a difficulty greater than the previous one (mastery learning). However, if the learners' results are negative the ITS can propose, for instance, a task with a lower difficulty or alternative learning content. Moreover, there is an inner loop for each task where a sequence of *steps* have to be executed by the learners in order to achieve the task objectives and provide a solution for the associated problem. ITS can provide adaptive feedback (positive, negative, etc.) in response to the learners' answers for the current steps or hints to anticipate the next step of the same task. The set of all possible actions that can be carried out by the ITS is called *tutoring actions*. The selection of an effective tutoring action can be accomplished on the basis of pedagogical strategies, learners' profiles, context, domain and so on.

2.2.2 Game-based Learning Environments

A Game-based Learning Environment is mainly characterized by the capability to maintain constant players/learners' attention and to keep high levels of engagement for them. One of the main challenges of the educational games is to provide a game design enabling players/learners to face tasks and steps according to their real ability (Kiili, 2005). The main idea in Game-based Learning is to always guarantee a game level a bit greater than the ability she demonstrated by considering a *zone of proximal development* (Vygotsky et al., 2012). This aspect can be sustained by the behaviour of an ITS by controlling and adapting both internal and external loops.

3 DEFINITION OF ITSEGO

The core part of this work is the definition of an ontology that can be used as a model to construct ITS for Game-based Environments. Such ontology is called ITSEGO. ITSEGO can be adopted, extended, instantiated and completed (with additional content) to build specific games empowered by intelligent tutoring features. In order to define ITSEGO we have selected the Web Ontology Language (OWL), provided by W3C[2], that is one of the main components of the so called Semantic Web Stack[3]. OWL is based on Description Logic (Krötzsch et al., 2014) and provides a set of suitable capabilities like providing high levels of interoperability and integration and a formal inference mechanism. This is one of the main benefits of Semantic Web ontologies, they make easily to integrate with existing results like, for instance, the ontology provided by the OMNIBUS ontology (Bourdeau et al., 2007).

3.1 Building the Ontology

In this section we will show the most relevant generic parts of the ITSEGO that are identified with respect to the important aspects of the Game-based ITS we are modeling: interactions, tasks and steps, context, storytelling, game, tutoring actions and tutoring rules. The main design principles considered for the engineering of ITSEGO are the following ones:

- Maintaining simple and general the design of classes, properties and axioms with no specialization with respect to application domain, learning context and content to propose. The idea is to provide essential building blocks (with basic constructs and key abstractions) for developing different ITS in several and heterogeneous domains, although the driver for this work comes from a specific target with specific motivations.
- Including, in the ontology definition, all the necessary to construct a specific (Game-based) ITS to be deployed at a testing environment (e.g. Protégé[4]) as well as to support real-world ITS.
- Designing and including features for building Game-based ITSs deployable in blended environments by considering game and storytelling elements.

3.1.1 Interactions

An interaction with the Game-based ITS is divided into sessions. A single play is an individual of class `ITSEGO:Play` and aggregates one or more sessions. The concept of session is modelled by using the class `ITSEGO:Session`. Each session maintains a set of interactions among learner (student) and system. Interactions are individuals of the class `ITSEGO:Interaction`. The learner is represented by means of the `foaf:Person` class included in the well known FOAF ontology[5] (see Fig. 1). An interaction refers to a task (individual of class `ITSEGO:Task`) and, in particular, to a step (individual of class `ITSEGO:Step`). Tasks and steps are mainly related by the `ITSEGO:isComposedBy` property. The property `ITSEGO:firstStep` indicates the first step in a task and the property `ITSEGO:next` provides an ordered relation among the steps within a task.

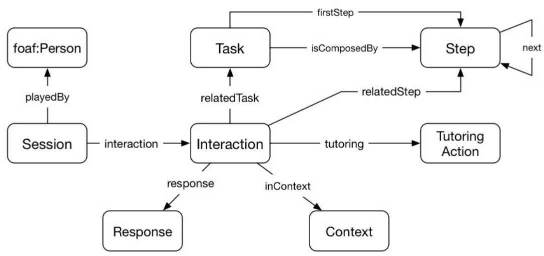

Figure 1: The ontology part dealing with interactions.

Tasks have difficulties (`ITSEGO:difficulty` data property) and can be related each others by means of the `ITSEGO:similarTo` property. This is useful to propose alternative tasks. Steps are related each others by means of the `ITSEGO:equivalentTo` property that is useful to accept multiple solutions for a specific step. Detailed information on tasks and steps will be provided in the next sections. An interaction also refers to a response (individual of `ITSEGO:Response`) that represents data related to the answer that the learner provides when a specific step in a specific task is proposed to her. An interaction occurs in a given context (i.e., a set of contextual information like, for instance, the learner' profile). Moreover, the interaction includes a tutoring action (individual of class `ITSEGO:TutoringAction`) that is the action the tutor (human or automatic) has to perform in order to adapt the experience by taking into account the specific learner's answer. When a learner's answer is evaluated, it is needed to check if it is correct. Some implementations could foresee also the chance to have, for instance, a fuzzy correctness value. In order to support different kinds of metrics, ITSEGO provides the class `ITSEGO:EvaluationResult` that is linked to and provides information related to a given interaction. This class can be specialized to support boolean evaluation results, fuzzy evaluation results and additional properties that can be possibly used in the pedagogical rules.

[2]http://www.w3.org/TR/owl2-overview/
[3]http://www.w3.org/standards/semanticweb/
[4]http://protege.stanford.edu/
[5]http://www.foaf-project.org

3.1.2 Tasks and Steps

The problems proposed to the learners are structured in tasks and steps. Classes `ITSEGO:Task` and `ITSEGO:Step` must also contain all the information needed to allow the execution of the Game-based ITS, i.e., how the problem is presented (communicated) to the learners. The base ontology we propose brings with it a set of properties useful to link tasks and steps with concrete resources (by using URIs). In this way it is possible to deploy the system at different and heterogeneous learning environments.

Furthermore, the `ITSEGO:Interaction` class provides a link to the specific environment (platform) used to deploy tasks and steps, receive user inputs and provide outputs.

Now, let us focus on two important aspects: topic and content of a task/step. Tasks and steps need to be associated to the content that must be proposed to learners by using generic URIs. In addition, ITSEGO provides a conceptual layer by means of the property `ITSEGO:topic` connecting `ITSEGO:Task` (and `ITSEGO:Step`) to `skos:Concept` that is the main construct of SKOS[6]. SKOS is a Semantic Web ontology allowing the definition of thesauri, taxonomies, concept maps and so on. The conceptual layer enables reasoning on topics and allows, for instance, to classify tasks with respect to their topics.

Lastly, tasks are not pre-ordered. The sequence of tasks, proposed to the learner, can be obtained by applying specific pedagogical rules. One of the plausible solutions is to adopt the *mastery learning* strategy, i.e., proposing tasks in order of increasing difficulties.

3.1.3 Context

The class `ITSEGO:Context` includes: student model information and contextual information like, for instance, environment or situation characteristics during learners' interactions with the system. The student model is composed of three parts. The first one includes student's information (e.g., name, age), contextual information (e.g., family context, school context), competency information (e.g., competencies already acquired), personal traits information. The second one is characterized by information related to a specific play like rewards (earned during the play), competencies (acquired during the play), score (the total score for the play) and performances (produced during the play). The performances of a learner are linked to specific game levels. Each game level has a status (completed, not-completed, not-started) and includes scores, rewards (that can be also badges) and competencies (acquired) obtained by the player when she faces this level. A game level is related to one or more tasks (possibly belonging to the same difficulty) that have to be successfully executed to complete the level. The third one is characterized by information related to affective and emotional states of the player/learner. This information is dynamic and can be "perceived" by processing raw data that comes from additional sensors.

All the information included in the context can be used by the pedagogical rules to adapt the experiences and produce suitable and effective tutoring actions. Moreover, some other data have to be gathered in addition to the previously indicated ones. For instance, pedagogical rules can be based on number of errors and number of consecutive errors for a specific step and so on.

3.1.4 Tutoring Actions

Tutoring actions are the actions provided by tutors in response to learners' interaction with the content of a specific step in a given task. Tutoring actions are selected by considering several and heterogeneous aspects: context, pedagogical strategy, student's emotional state, student's prior knowledge and so on. Tutoring actions can be classified in feedback, hints and adaptations. ITSEGO provides further specializations of the above mentioned classes. At each interaction, only a subset of all plausible tutoring actions can be provided.

3.1.5 Tutoring Rules

Tutoring or pedagogical rules implement the pedagogical strategy used to adapt the environment and generate feedback/hint on the basis of the behavior of the learner during her interaction with a step (in a task) in a specific context. There are numerous ways to implement such rules for an ITS. In this work, we propose the development of such rules as *class restrictions* within ITSEGO. In this way it is possible to generate the correct tutoring action (or set of tutoring actions) by using standard OWL-DL reasoners.

The part of the ontology that is useful to model and execute pedagogical rules is provided in Fig. 3. The class `ITSEGO:PedagogicalRule` contains all the admissible rules of the ITS (individuals). Each rule is composed by a condition and an action (or more actions) that must be executed when the condition value is *true*. Conditions are individuals belonging to `ITSEGO:LearningCondition` and are composed by game state (task and step executed by the learner), context (up to date information related to the

[6] http://www.w3.org/2004/02/skos/

learner: scores, attempts, emotions, profile, etc.), response (correctness and other information about the learner's response with respect to the considered task and step) and story environment (the configuration of the story environment underlying the game). Such elements are considered in the condition in order to make decisions about the tutoring action to deliver. The idea is that if the current learning condition is equal (for each element) to the learning condition attached to a rule, such rule can be executed to obtain the right tutoring action. Furthermore, `ITSEGO:ExpectedResponse` is linked to the class `ITSEGO:Step` and indicates response templates for a step. Each response template (for a step) contains also a score. In this way it is possible to evaluate the current response of the learner by finding the best match among the current response and the existing response templates. Tutoring actions are individuals of class `ITSEGO:TutoringAction` and are connected to the rules by means of the property `ITSEGO:isImplied` (and its inverse `ITSEGO:hasAction`).

The selection of the correct tutoring action is obtained by looking for the rule generating such action. This selection is accomplished by considering two inference operations. The first one is used to look for the right pedagogical rule in the knowledge base by comparing the learning conditions associated to each existing rule and the current learning condition:

$$\begin{aligned} APR &\equiv (PedagogicalRule \sqcap C_1) \\ C_1 &\equiv (\exists hasCondition.(C_3 \sqcap C_4 \sqcap C_5)) \end{aligned} \tag{1}$$

$$\begin{aligned} C_3 &\equiv (\exists template.C_6) \\ C_6 &\equiv (\exists inverseMatchesWith.C_7) \\ C_7 &\equiv (\exists responseInCondition.CLC) \end{aligned} \tag{2}$$

$$\begin{aligned} C_4 &\equiv (\exists context.C_8) \\ C_8 &\equiv (\exists contextInCondition.CLC) \end{aligned} \tag{3}$$

$$\begin{aligned} C_5 &\equiv (\exists gameState.C_9) \\ C_9 &\equiv (\exists stateInCondition.CLC) \end{aligned} \tag{4}$$

The second one is used to extract the suitable tutoring action from the selected pedagogical rule:

$$TutoringAction \sqcap (\exists isImplied.APR) \tag{5}$$

The above described inference operations are graphically explained in Fig. 2 that shows how the current learning condition matches with the condition attached to a specific rule. In particular the individual `currentLearningCondition` (belonging to class `ITSEGO:CurrentLearningCondition`, in brief `ITSEGO:CLC`) matches with `learnincCondition01` because they have all equal elements and thus the right rule is `rule01`.

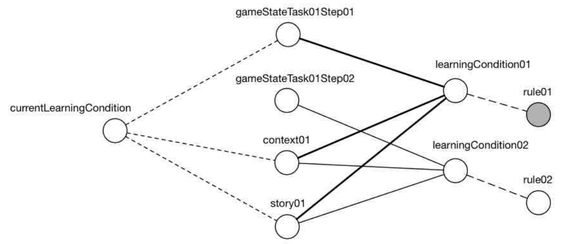

Figure 2: Sample individuals related to the operation of looking for the right rule to be applied.

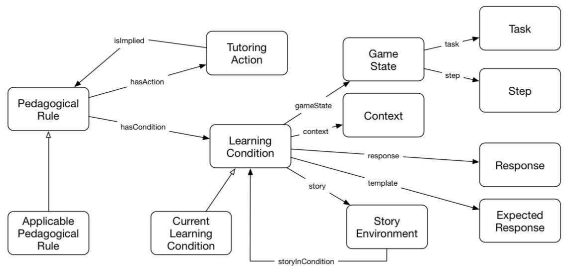

Figure 3: The ontology part dealing with tutoring (pedagogical) rules.

For the sake of clarity, and in order to provide simple statements, we have proposed only individuals like `context01` that includes several contextual information (the emotional state of the learner, the affective state of the learner and the tentative number for this step in this task, etc.). This does not imply a lack of generality because it is possible to simply provide more details by fragmenting individuals and considering, for instance, finer contextual informations. Take care that it is possible to implement pedagogical rules by means of other approaches enabling the definition of more complex rules. One of such approaches could be SWRL[7].

3.1.6 Storytelling

In order to model storytelling aspects within the ITSEGO, our approach is not providing "building blocks" defined from-scratch but integrating an existing language/ontology. In literature, several Ontologies and markup languages providing models to build stories are recognizable. Taking care our aim to maintain ITSEGO as light as possible we have selected a generic storytelling ontology model (Nakasone and Ishizuka, 2006) based on the Rhetorical Structure Theory (RST). We arbitrary assign the namespace `RST` to this ontology. In ITSEGO, a play is linked to the elements of

[7] http://www.w3.org/Submission/SWRL/

the story in which the play takes place. In particular, `ITSEGO:Play` is linked to `ITSEGO:Story` by means of the `ITSEGO:bgStory` property. A story is composed of story elements (individuals of `ITSEGO:StoryElement`). Story elements can be characters (individuals of `ITSEGO:Character`), settings (individuals of `ITSEGO:Setting`) and scenes (individuals of `ITSEGO:Scene` that is equivalent to `RST:Scene`). A scene is defined as a set of acts (individuals of class `RST:Act`) that are hierarchically composed of nucleuses and satellite entities linked by relations. Acts establish the minimum level of story organization. Such entities could contain an event (individual of class `RST:Event`) or another act. Agents are actors taking part in a scene by executing or being part of one or more events. Agents are individuals of class `RST:Agent` (that is equivalent to the class `ITSEGO:Character`). A role is a part that an agent plays during a scene. Roles are modelled by means of the class `RST:Role`. There are several roles: questioning role, informing role, contrasting role and evaluating role. ITSEGO adds to RST the class `ITSEGO:Setting` that provides the graphical elements of a scene.

3.1.7 Game

In order to embed an ITS in a game-based environment it is needed to enrich the ITSEGO with specific elements. In particular, the class `ITSEGO:Score`, that is linked to `ITSEGO:ExpectedResponse`, is used to model the score earned by successfully executing the related step. The `ITSEGO:Score` class can be linked also to other classes. For instance, if we link individuals of `ITSEGO:Interaction` with individuals of `ITSEGO:Score` it is possible to sign the score of each interaction. In addition, `ITSEGO:Score` can also be linked to several parts of the context (as reported in section 3.1.3) in order to assign scores to play and levels for each player/learner). Game levels can be modelled by means of the class `ITSEGO:GameLevel`, whose individuals are linked to the individuals belonging to `ITSEGO:Task`. The class `ITSEGO:GameLevel` maintains all the information needed to manage the game like, for instance, rewards (individuals of class `ITSEGO:Reward`) and max time to play (individuals of class `ITSEGO:Time`).

3.2 Use Cases for ITSEGO

In this section, we briefly describe three main use cases in which ITSEGO can be exploited.

UC#1 Sharing Terminology. ITSEGO can be used to share knowledge about terminology related to ITS among the community of researchers, system designers, software developers, content developers and educators. The adoption of the Semantic Web Stack fosters interoperability and integration.

UC#2 Building ITS. ITSEGO can be used to build ITS. ITSEGO can be used as-is for rapid prototyping by populating it and using the embedded pedagogical rules. Populating the ontology means to construct the domain model (by defining the concept map as indicated in Section 3.1.2 and linking those concepts to the new inserted tasks and steps), the expert model (by providing the expected response for each step), the tutor (pedagogical) model (by defining new pedagogical rules as *class restrictions* or logical rules), the student model (by filling learners' profiles) and, lastly, the communication (user interface) model (by configuring stories, characters, events, etc.). Moreover, ITSEGO can be also used within a real-world system developed by means of a general-purpose programming language like Java and some GUI framework. Lastly, ITSEGO can be extended (or specialized) by subclassing existing classes and adding new axioms and rules.

UC#3 Tracing and Learning Pedagogical Rules. ITSEGO can be used as a knowledge base for existing ITS. The `ITSEGO:Interaction` can be populated every time a learner interacts with a step. This allows to trace all actions occurring in the system to perform statistics, analytics and/or machine learning. In particular, machine learning algorithms can be used to learn pedagogical rules by using the (human) tutoring actions applied against a specific learner's interaction.

4 INSTANTIATING A GAME-BASED ITS

In this section we will show how the proposed ontology has been instantiated in order to build a tool supporting children to their transition from kindergarten to primary school. In particular, an Educational Game (in brief Edu Game) has been implemented in the form of an Android App. Lastly, a validation and evaluation framework is described in the second part of this section.

4.1 The Prototype System

This part of the work aims at introducing the most important issues related to the development of the prototype App that has been experimented.

4.1.1 Architectural and Technological Issues

The system prototype has been developed by means of LibGDX[8] and Java for realizing the front-end and deploying it on Android tablets (LibGDX is multi-platform, so it allows to deploy the software on Android, iOS, Desktop, Web, etc.). In this first prototype the access to ITSEGO is realized by means HTTP connections to a set of Web Services (implemented in Java and hosted by JBoss) to query and perform inferences over the knowledge base. The main idea of the system behaviour is that the App proposes a screen to the child in combination with the audio explanation (it is probably that 5-6 years old children cannot read text) of the associated task/step. At the end of the audio description, the child is free to interact with the screen. This interaction triggers an inference over ITSEGO that, according to the pedagogical rules modelled in the ontology, is able to find a suitable tutoring action to return it to the App that executes such action in order to provide its result (e.g. feedback) to the child. It it important to underline that the App implements exactly the ITSEGO model and its behavior mirrors the one provided by the pedagogical rules modelled by using the ontology constructs (as shown in section 3.1.5). Pedagogical rules, added to the ITSEGO, are driven by the specific context (5-6 years old children) in which they have to be used and for the specific learning objectives (supporting the transition from the kindergarten to the primary school) we have to accomplish.

4.1.2 Edu Game Content

In order to design the Edu Game content, we have selected a set of characters, one for each task, who guide the child along the steps. Each task is associated to a story providing a problem to the child who has to solve it by executing a sequence of steps. In particular, the first task is associated to a specific problem of *tidying up a room*. The main character for this task is called *Rughetto* and he asks to the child/player to give him help in order to allow him to go out to play football with his friends (see Fig. 4(a)). Each step associated to this task is related to the main problem. For instance, in the first step the child/player has to put toys into the basket games and the colored pencils into the case (i.e. a problem related to classify things). In the second step the child/player has to sort some books from the smallest to the largest (i.e. a problem related to the concepts of smaller and larger things). The third step is focused on the identification of the right shapes and the fourth (last) step is focused on

[8]https://libgdx.badlogicgames.com

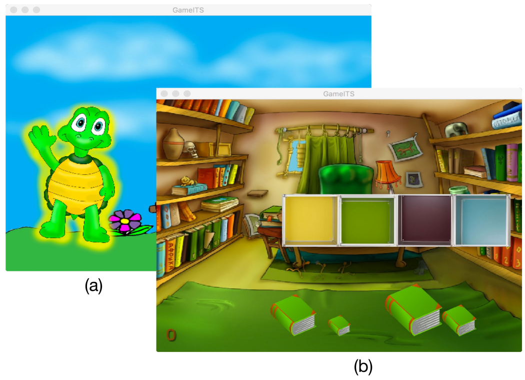

Figure 4: Screens: (a) introduction and (b) second step.

finding correct numbers on the phone keypad in order to call *Rughetto*'s friends to play football.

Feedback actions are driven by pedagogical rules and are based on audio descriptions and colored screens. Hint actions are enriched by audio helps and/or images recalling suitable solutions for specific steps. Colors are selected with respect to the emotions that should be encouraged.

4.2 Validation and Evaluation

In order to evaluate and validate the results of this work we have designed the activities of Table 1.

Table 1: Framework for Validation and Evaluation.

	Validation	Evaluation
ITSEGO	*OOPS! Methodology*	*Case Study*
Ontology-driven Approach	*Building of Edu Game "Rughetto"*	*Asking to Developers*
Edu Game "Rughetto"	*Testing*	*Experiments*

ITSEGO. The ontology ITSEGO has been validated by means of the OOPS! tool (Poveda-Villalón et al., 2012) for detecting potential pitfalls that could lead to modelling errors. In particular, ITSEGO has passed the tests with respect to *Functional Dimension*, *Consistency* and *Conciseness*. The other tests for *Structural Dimension* and *Completeness* advise that there is only one important pitfall. In particular, pitfall P11 (missing domain or range in properties) is signaled for some properties defined as inverse of other properties. Thus, this pitfall is fixed when reasoning is executed on ITSEGO and missing domains and ranges are inferred for such properties. Lastly, ITSEGO has been evaluated by populating it with real

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

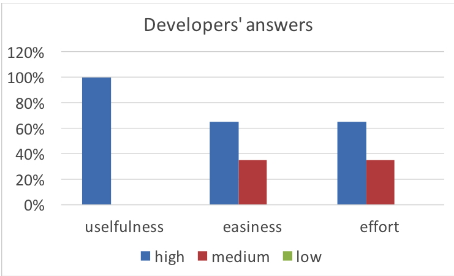

Figure 5: Results of developers' questionnaires.

data for the case study that has been designed, implemented and described in Section 4. The Edu Game, exploiting ITSEGO for generating tutor actions, has been tested and its behavior produces correct results.

Ontology-driven Approach. The approach we proposed has been validated by building the Edu Game described in Section 4. This approach has been evaluated by proposing a questionnaire to the three developers of the Edu Game and analysing their answers. Such questionnaire includes 9 questions regarding three aspects of the approach: i) usefulness of the shared terminology formalized by ITSEGO, ii) easiness of implementing pedagogical rules, and iii) effort of building contents. Results are reported in Fig. 5 that shows the usefulness and easiness of the interaction with ITSEGO. Difficulties and high efforts to build Game-based ITS contents are emphasized.

Edu Game. The Edu Game has been validated by considering both unit testing and system testing. An experimentation activity has been planned to evaluate the effectiveness of the learning process realized by means of the provided Edu Game. More in details, the experiment has been designed to be performed in the context of the first class of a primary school. Three groups will be built. The first two groups are control groups. The third group is the experimental group. In the first phase a pre-test will be dispensed to the children of the three groups. In the second phase, the children of the third group will interact with the game in order to learn the subjects introduced in Section 4. The first two groups will deal with the same subjects by using traditional learning content. Lastly, in the third phase, all the groups will face the same post-test.

5 CONCLUSIONS

This paper proposes the idea of integrating ITS features into Edu Games. Main results are: definition of an ITS ontology, definition of an ontology-driven approach to build Game-based ITS and development of an Android App implementing an Edu Game that exploits the reasoning capabilities of ITSEGO. Lastly, a framework for validating and evaluating the multiple results of this work has been proposed. In the future, the deploy of ITSEGO on Raspberry will be investigated. At the moment we are realizing the experiment described in the last part of Section 4.

REFERENCES

Bourdeau, J., Mizoguchi, R., Hayashi, Y., Psyche, V., and Nkambou, R. (2007). When the domain of the ontology is education. *Proc. of I2LOR07*.

Collins, A. M. and Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407.

du Boulay, B. (2006). Commentary on kurt vanlehn's" the behavior of tutoring systems". *IJ Artificial Intelligence in Education*, 16(3):267–270.

Jackson, G. T. and McNamara, D. S. (2013). Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology*, 105(4):1036.

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and higher education*, 8(1):13–24.

Krötzsch, M., Simancik, F., and Horrocks, I. (2014). Description logics. *IEEE Intelligent Systems*, 29(1):12–19.

McNamara, D. S., Jackson, G. T., and Graesser, A. (2010). Intelligent tutoring and games (itag). *Gaming for classroom-based learning: Digital role-playing as a motivator of study*, pages 44–65.

Mizoguchi, R., Vanwelkenhuysen, J., and Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, pages 46–59.

Nakasone, A. and Ishizuka, M. (2006). Storytelling ontology model using rst. In *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 163–169.

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

Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012). Validating ontologies with oops! In *Knowledge Engineering and Knowledge Management*, pages 267–281. Springer.

Vygotsky, L., Hanfmann, E., and Vakar, G. (2012). *Thought and language*. MIT press.