An Intelligent Tutoring System for Procedural Training with Natural Language Interaction

José Paladines[1,2] and Jaime Ramírez[2]
[1]*Facultad de Ciencias Técnicas, Universidad Estatal del Sur de Manabí, Manabí, Ecuador*
[2]*Computer Science School, Universidad Politécnica de Madrid, Madrid, Spain*

Keywords: Context-aware Dialogue, Intelligent Tutoring System, Natural Language Processing, Procedural Training, Virtual Environment.

Abstract: In this paper we present a proposal of an Intelligent Tutoring Systems equipped with dialogue in natural language to facilitate student interaction with the learning environment, provide hints and answer students' questions. This system is designed to be integrated with a 2D/3D virtual environment for procedural training, where it can maintain a dialogue with students adapted to the context. Our notion of context comprises: the specific features of the student; his/her progress in the development of the task; and the virtual environment where it is performed. The dialogue will be controlled by a dialogue manager, built on Watson Assistant, which has been chosen for its versatility. Additionally, we present an application example that describes the operation of the modules that constitute the proposed approach. Then, we provide some indications on how it will be evaluated with students shortly.

1 INTRODUCTION

During the last years, Intelligent Tutoring Systems (ITS) have been developed that show effective results in the teaching the concepts of physics, mathematics and computer science. Some of these ITSs have been able to capture the attention and interest of most students through mixed conversational dialogues and have been able to produce significant learning gains beyond the classroom environment (Graesser *et al.*, 2001).

Furthermore, 2D/3D Virtual Environments (VEs) have demonstrated to be valuable training tools by simulating real scenarios where students can learn without risks in a cost-efficient way (Dalgarno, 2002; Duncan, Miller and Shangyi, 2012). 3D VEs offer a superior learning experience, greater immersion, greater fidelity and a high level of active student participation, therefore, in addition to supporting the learning tasks, can be intrinsically motivating to the student to make decisions to achieve individual goals within the environment (Dalgarno and Lee, 2010). This kind of virtual environments become especially useful for procedural training, that is, when students have to learn to perform a task step by step that eventually they will have to do in the real world.

However, after carrying out a literature review in the field of ITSs, we can state that there are some ITSs intended for procedural training, in which the student can interact with a 2D/3D virtual environment, such as Steve (develop physical tasks) (Rickel and Johnson, 1999), SafeChild (pedestrian safety of children) (Gu, Sosnovsky and Ullrich, 2015), Lahystotrain (surgeons in laparoscopic operations and hysteroscopy) (Los Arcos *et al.*, 2000), TRANSoM (pilots of remote-operated submarine vehicles (ROV)) (Pioch, Roberts and Zeltzer, 1997), but only in a few of these ITSs (Jacob, Paco and Normit-SE) the user can interact with the system through a dialogue in natural language.

The goal of this paper is to present a proposal for an ITS capable of maintaining a dialogue with the student in natural language while the student is performing a procedural training in a 2D/3D virtual environment. To generate the dialogue in natural language, the specific features of the student, his/her progress in the development of the task and the physical environment where it is performed will be taken into account. Therefore, the tutoring feedback will include a dialogue adapted to the context.

Moreover, in a virtual environment for procedural training, the actions related to the development of a procedure become particularly relevant and must

Paladines, J. and Ramírez, J.
An Intelligent Tutoring System for Procedural Training with Natural Language Interaction.
DOI: 10.5220/0007712203070314
In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)*, pages 307-314
ISBN: 978-989-758-367-4

serve to determine the most suitable tutoring at each moment.

To generate the dialogue between the student and the ITS, we will build a dialogue manager with the cognitive services of some currently available platforms (Mallios and Bourbakis, 2016).

This paper is structured as follows: section 2 presents the previous works most related to the objectives of this paper; Section 3 summarizes the main features of the platforms for the construction of dialogue managers; Section 4 details the architecture of the proposed system; Section 5 describes, through an application example, the operation of the ITS integrated with the dialogue manager; and finally, section 6 shows the conclusions and future work.

2 RELATED WORK

In the literature we find few ITSs that use dialogue in natural language for procedural training, because most of this kind of ITSs are oriented to the teaching of concepts. Therefore, we believe that to show a reasonable view of the state of the art in this field, it is convenient to first present some remarkable systems that simulate patterns of discourse and include pedagogical strategies of a human tutor through dialogues for declarative instruction, and then present the existing ITSs with dialogue for procedural training.

CircSim (Glass, 2001), considered one of the first systems to implement dialogues in natural language, can analyse the user input through a syntactic analysis and admits short answers of one or two words. It has a tutoring strategy based on Directed Reasoning Lines to control the variables of a prediction table and is oriented to the physiology domain. Atlas-Andes (Rosé *et al.*, 2001) adds dialogue capabilities to the Andes system (Gertner and VanLehn, 2000), allowing students to participate in a typed dialogue. It has a tutoring strategy based on knowledge construction dialogues (KCD) by means of which it involves the students in a dialogue about correcting conceptual misconceptions of physics. Why2-Atlas (VanLehn *et al.*, 2002) encourages students to write essays as answers to a question. The analysis of these long explanations and the discovery of misconceptions is possible through CARMEL. Like Atlas-Andes, it is aimed at the teaching of physics. ITSpoke (Litman and Silliman, 2004) is a system that involves students in spoken dialogues of conceptual physics. To analysis the statements of students, it uses the front-end of the Why2-Atlas system. Regarding the tutoring strategy, only Atlas-Andes, Why2-Atlas, ITSpoke ITSs rely on KCDs. These KCDs are based on the CircSim Directed Reasoning Lines. Other outstanding ITS for declarative instruction is AutoTutor (Graesser *et al.*, 2005), a system that maintains conversations of mixed initiative with students to allow them to build explanations of concepts. It uses a dialogue pattern called expectation and misconceptions tailored dialogue that consists in comparing the explanations of the students with a set of expectations (ideal answers) and misconceptions (incorrect answers) using a statistical technique called latent semantic analysis. This system has evolved over time and has managed to cover different domains such as computer literacy, biology, physics and critical thinking. Beetle II (Dzikovska *et al.*, 2014), implements an approach based on a task-based dialogue system supported by a simulation that generates a dynamic learning context. The dialogue with the student is executed through a cycle "Predicting, verifying, evaluating" with which we can analyse the predictions and mistakes that the student may make. It is aimed at teaching electricity and electronics and uses an ontology to represent the knowledge domain. Both AutoTutor and Beetle II support the constructions of explanations through the dialogue but differ in the approach to interpret the student's statement. AutoTutor applies a statistical approach whereas Beetle II employs a hybrid approach, that is, adds a statistical classifier to the semantic analyser for a better interpretation of statements.

Next, we will mention the only ITSs with dialogue in natural language for procedural training. Jacob (Evers and Nijholt, 2000) teaches how to solve the problem of the Towers of Hanoi and provides instructions and assistance to execute tasks in a virtual 3D environment. Paco (Rickel *et al.*, 2002) supports tutoring actions as part of a collaborative dialogue system (built on collagen) that uses rules for the generation of speech acts. This generation of acts is based on a task model, a student model and the interaction with the student. Normit-SE (Mitrovic, 2005) teaches the database normalization process. The dialogue starts from the moment the student makes an error to which he must provide explanations by selecting one of the options (solutions) offered in a menu. These systems provide help when requested and positive comments, although in the case of Jacob it is occasional. To analyse the students' statements, they use a symbolic approach based on superficial semantic grammars.

3 DIALOGUE MANAGER

Nowadays, there are some platforms on the market to implement conversational interfaces; among the most popular are Google's Dialog Flow, Facebook Messenger's Wit.ai, IBM's Watson, Microsoft's LUIS, etc. Each of them has its own characteristics, advantages and disadvantages, but to structure the flow of conversation or dialogue, these platforms use some common elements such as utterances, intentions and entities, which facilitate the Natural Language Processing (Singh, 2017).

- An utterance is the user input that the application needs to interpret, which in some cases are not well formulated.
- An intention represents the purpose expressed in the user's input. The same intention can be expressed through different user sentences.
- An entity represents an instance of an object class that is relevant to a user's intention and can be identified in an utterance.

Something outstanding of these platforms is that they support the use of a structure called context that facilitates the adaptation of the dialogue to different situations in a given scenario. This structure is used internally and externally to pass information between a client application and the dialogue manager.

For this work, we have only considered the IBM, Microsoft and Google platforms because apart from the dialogue managers services, they also have a natural language understanding module necessary to decompose the student's statement into entities and relationships. However, after comparing these three platforms, we have chosen IBM because it enables us, among other things, to handle a larger number of intents; to define concepts, dictionaries and relationships through annotators; and to manage the context information more easily. In addition a recent comparative study of conversational platforms (Koplowitz *et al.*, 2018) positions the IBM platform as the most complete and robust of the market.

From the IBM Watson platform, the following cloud services will be used:

- Watson Knowledge Studio (WKS), to create the automatic learning annotator by identifying the mentions (entities) and relationships in unstructured texts;
- The Natural Language Understanding (NLU) module (IBM, 2016) to apply the machine learning annotator obtained from the WKS; and,
- The Watson Assistant (IBM, 2018) to build the nodes of dialogue based on the intents, entities and context variables necessary for the training.

4 PROPOSED APPROACH

The dialogue manager associated with the ITS must have enough information to be able to provide a contextualized dialogue as part of the tutoring feedback. To provide this type of dialogue, the context must contain information related to the student's knowledge, the virtual environment (with static and dynamic information) and the student's current progress in the practice assignment to be performed. Some examples of personalized tutoring feedback that may be provided in this way are the following ones:

- Answer questions related the ubication and identification of an object in the virtual environment or how reach it, even though this object is a distant position to the student's avatar one
- Answer questions related the next action to be done in the practice assignment.
- Recommend learning activities to fill gaps in knowledge demonstrated by the student.
- Provide hints proactively to guide the student with the execution of a task, if it is observed that the student needs help, even if he/she is not asking for it.
- Encourage an affective dialogue to mitigate students' inactivity or moments of discouragement.

Figure 1 describes the architecture of the ITS with dialogue in natural language. This architecture contains four main components: the Procedural Training Environment (PTE), the Natural Language Understanding (NLU) System, the ITS and the Dialogue Manager (DM).

The PTE is the module that simulates the real environment where the task related to the practice are carried out. In order to give more realism to the tasks that the student must perform throughout the practice, this environment can be a virtual world in 3D. The interaction of the user with the PTE can be generated through events such as questions, attempts of actions, etc., that will be delivered to the ITS during the development of the practice.

The NLU System will be responsible of receiving the user's sentences and extracting from them their composing entities and relationships. In this way, it will preprocess the statement of the user, so that later the ITS modules can work with the semantics of the statement. The NLU system will be developed using the Watson Knowledge Studio and the Watson's Natural Language Understanding Service.

The ITS will be integrated by the modules corresponding to a classic ITS plus a world module. The World Module (WM) will represent the physical characteristics of the virtual training environment, that is, it will contain information about the scenarios and constituent aspects of the virtual Its content will be useful so that the system can answer questions, for example, about the situation of an object or how to go from one place to another. environment, related to avatars and 2D/3D objects. The Student Module (SM) will contain information related to the student. In this work, we will adopt the student model proposed by (Clemente, Ramírez and de Antonio, 2011), because it fits very well to your needs. This SM contains information on: the student's actions; his/her movements through the virtual environment; the questions he/she asked; the hints he/she received from the tutor, etc. From this information, the same module will infer, with a certain level of reliability, the student knowledge that, in turn, will be useful to decide the best tutoring strategy in each moment. Both the information of the WM and the SM will be represented by means of ontologies, because they support the representation of sufficiently abstract and properties as well as facilitate their own reuse and even their own extension to other application contexts, if necessary. To access the information on these ontologies, Jena framework will be used. The Expert Module (EM) will contain information about the knowledge that the student must learn. In this case, the EM will contain a complete description of the procedure to be learned. The Tutor Module (TM), based on the information of the other modules, will provide students with adequate feedback at every moment of their learning. The ITS extracts the information from the different modules to build the context. This context will be represented by an ontology and will be filled with information from the SM (state of knowledge, progress of the activity, student's trajectory in the virtual environment), the EM (the correct plan, the next correct action) and the WM (structure of the virtual world, position of the student's avatar, descriptions of the objects). Once the contextual information is collected, before passing it to the Dialogue Manager, this information will be transformed into another representation understandable by the Dialogue Manager.

The DM will contain the definition of the structure of the dialogue, i.e., the intentions, the entities and the dialogue nodes specifically intended to the training environment. This component will be implemented through the Watson Assistant and will be responsible for the dialogue with the user taking into account the contextual information provided by the ITS.

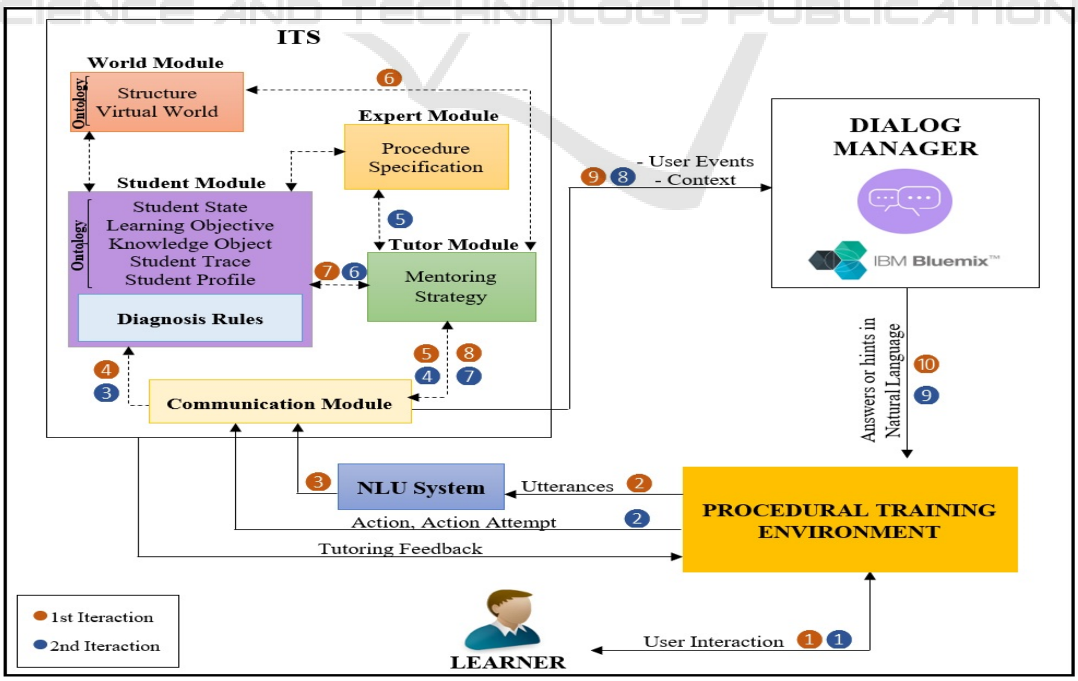

Figure 1: Architecture of proposed approach.

5 EXAMPLE OF APPLICATION

For a better understanding of the tutoring process and the generation of dialogue in natural language, we present an example taken from a practice assignment that is performed in a Virtual Biotechnology Laboratory (In http://youtu.be/ mAFREZ5_iak you can find a video of this virtual laboratory). This laboratory was developed on the platform of virtual worlds OpenSimulator, as part of a master thesis (Riofrío-Luzcando, 2012). To carry out the practice assignment, the student controls an avatar and has he help of an automatic tutor, who will give him indications about the actions to be carried out at each moment and will show him error messages, when he makes a mistake. These indications and error messages consist of messages previously configured and associated with each of the actions that must be carried out in the practice assignment.

The current version of the automatic tutor doesn't implement the architecture proposed in the previous section. However, the example that will be explained below describes the behavior of the ITS as if it were really equipped with the proposed architecture and implemented the process of tutoring detailed in the previous section.

In this example, we are going to assume that the student asks for a chemical that he needs to prepare the mix, and after receiving the hint, he looks for it and ends up adding a different chemical to the one he is looking for. The tutor will give the hint based on the student's level of knowledge.

STUDENT (Utterance) Where is Casein? / Where can I find casein?
TUTOR (General hint) "Casein is in the showcase"
STUDENT (Action attempt) Look in the showcase and try to add the bisacrylamide chemical.
The tutor detects that the student has taken an element that isn't casein and decides to block the action, send an error message and give a more specific hint about the chemical to choose.
TUTOR Error Message "The action attempt is incorrect"
(Specific hint) "You must add casein to the mix, that's the right thing!

As we can see, the dialogue is composed of two iterations; and to show how the tutoring feedback is generated, Figure 1 has been used as a base, so that enumerated circles have been drawn on it to specify the sequence of steps executed by the ITS modules.

Table 1: Iterations of the Dialogue Example.

First Iteration: The student asks the ITS, because he/he does not know where Casein is.
1. The student asks where Casein is 2. The PTE retransmits the question to the NLU System. 3. The NLU System interprets the question and breaks it down into entities and relationships and sends them to the ITS Communication Module (CM). 4. The CM sends the question broken into entities and relationships to the SM. • The SM infers that the student doesn't know where the casein is and updates the student's knowledge state in the ontology. • The learning objective to be evaluated is: "The student knows where the casein is". • Therefore, the state of knowledge of the student will contain "The student has not acquired that objective with a certain degree of certainty". 5. The CM sends the question to the TM Where is Casein? 6. The TM, once the student's question arrives, asks the WM Where is casein? Where is the student? Then, WM returns: "Casein is in the showcase" and "The showcase is in the main room" "The student is in the main room" 7. The TM asks the SM: Does the student know where the showcase is? What was the overall performance of the student so far in practice assignment? As the student has previously taken another chemical from the showcase and this was recorded in the student's ontology, the SM will answer the TM: "Yes, the student already knows where the showcase is" "The performance was good (made a few mistakes)" 8. The TM sends to the CM the information obtained from the WM and SM with which the context information will be elaborated. In addition, it indicates that the student is going to need a general hint. 9. The CM sends the student's question to the DM. Where is Casein? and the context with the following information: "The student is in the main room". "Casein is in the showcase". "The showcase is in the main room". "The student knows where the showcase is". **Hint Level:** General 10. The DM builds the hint according to the level decided by the TM and send it to the PTE. **General:** "Casein is in the showcase"
Second Iteration: After receiving the hint, the student tries to add bisacrylamide to the mix, so the action attempt is blocked.
1. The student tries to add the bisacrylamide chemical element to the mix. 2. The PTE retransmits the action attempt (event) to the CM.

3. The CM sends the event to the SM.
   - The SM registers in its ontology that the student tries to perform an action.
   - The learning objective to be evaluated is: "The student must add casein to the mix".
4. The CM sends the event to the TM.
5. The TM asks the EM if the event is correct and what is the next action; and EM returns: "The event is wrong", "Add casein to the mix".
6. The TM tells the SM "The action of adding bisacrylamide to the mix is incorrect" and asks What was the recent performance of the student so far in practice assignment?
   - The SM infers that the student has not acquired this learning objective or reinforces his belief that he has not acquired it, since he has just obtained new evidence of it.
   - The SM returns that the performance was bad for the next action.
7. The TM tells the CM:
   "The action of adding bisacrylamide to the mix is incorrect".
   NEXT_ACT_PLAN: "Add casein to the mix".
   In addition, it indicates that the student is going to need a **specific hint**.
8. The CM tells the DM the following.
   "The attempt of action is incorrect".
   NEXT_ACT_PLAN: "Add casein to the mix".
9. The DM sends the PTE an error message and hint according to the level decided by the TM.
   "The attempt of action is incorrect".

Specific: "You must add casein to the mix, that's the right thing!"

5.1 Application of the Natural Language Understanding

The Natural Language Understanding system has an automatic learning annotator that was built from some entities, dictionaries and an annotated corpus of dialogue. These entities, dictionaries and the corpus refer to the terminology associated with the procedural tasks that students must perform as part of the practice assignment. If we apply this annotator to the student's question, the annotator will generate a file in JSON format that would have the following information in abbreviated form:

Utterance: Where is casein?/ Where can I find casein?
Relation: LocateObj
Entities: Adv_ans (Where),
Action (Is),
Chemical (Casein).

5.2 Application of the Student Module

The Student Module is responsible for controlling the information related to the learning objectives, the state of the student knowledge, the trace of the student, etc. In this sense, this information is represented by a network of ontologies and updated by means of diagnosis rules (Clemente, Ramírez and de Antonio, 2011).

First Iteration:
Learning objective: "The student knows where the casein is"
State of the student's knowledge: Student does not know where the casein is.
Rule:
Type_Of_Question(question,Where_Is(obj)) →
Add_SM(¬Know(Where_Is(obj)))

Second Iteration:
Learning objective: "The student must add the casein to the mixture"
Trace of the student: Student tried to add bisacrylamide.
Rule:
Try_To_Apply(actx) ∧ Next_Act_Plan(acty) ∧
¬Equal(actx, acty) →
Add_SM(¬Know(Next_Act_Plan(acty)))

5.3 Application of the World Module

Within the World Module there is an ontology, expressed in OWL language, in which information about the 3D virtual environment has been represented, that is, the names and locations of the contained objects, the location of the student's avatar, etc. In this case, this module provides the following information:

First Iteration:
```
<owl:ObjectProperty rdf:ID="Is">
<rdfs:domain
rdf:resource="#chemical"/>
<rdfs:range
rdf:resource="#container"/>
</owl:ObjectProperty>
```

Second Iteration:
```
<owl:ObjectProperty rdf:ID="Is">
<rdfs:domain
rdf:resource="#container"/>
<rdfs:range rdf:resource="#room"/>
</owl:ObjectProperty>
```

5.4 Application of the Tutor Module

The Tutor Module decides and applies the tutoring strategy according to the pace of the student's learning, just as a human tutor would. To do this, it interprets the requests of the communication module; collects the information coming from the world, expert and student modules; and encapsulates the necessary logic to return the information required by the Dialogue Manager, and thus enables the dialogue with the student. To build this information in the example, TM will have to submit the following requests:

First Iteration, TM requests information to:
WM: Where is the casein? Where is the student?
SM: Does the student know where the showcase is? What was the overall performance of the student so far?

Second Iteration, TM requests information to:
EM: Is this event correct or incorrect?
SM: What was the recent performance of the student?

5.5 Application of the Dialogue Manager

The Dialogue Manager receives from the CM the user's event and the context data. Then, with this information the dialogue manager generates the response in natural language. For the generation of natural language, a dialogue structure has been designed, integrated by the necessary intents, entities and dialogue nodes. Next, the dialogue structure and the contexts employed in the example are detailed in a simplified form:

First Iteration
Intent #locate = Where is Casein?
Entities @chemical = Casein

Context variables
$levHin: “G”,
$posStu: “main room”,
$objLoc: “casein”,
$contObjLoc: “showcase”,
$ubiSpaObj: “in”,
$posObjLoc: “main room”,

Node information
If #locate and (@element || @document || @chemical)
If ($levHin== “G”) && ($posStu == “main room”)

The answer in natural dialogue would be:
The $objLoc is $ubSpaObj the $contObjLoc

Second Iteration
Intent #trylocate = “ ”
Entities @chemical = “bisacrylamide”

Context variables
$levHin: “C”,
$posStu: “main room”,
$objLoc: “casein”,
$contObjLoc: “showcase”,
$posObjLoc: “main room”,
$nextAccPlan: “Add casein to the mix”

Node information
If #trylocate and ($posStu == “main room”) and ($objLoc!=@chemical)
If ($levHin== “C”)

The answer in natural dialogue would be:
“The attempt of action is incorrect”
“You must $nextAccPlan, that’s the right thing!”

6 CONCLUSION AND FUTURE WORK

We have presented a proposal of an Intelligent Tutoring System for Procedural Training with Natural Language Interaction. To generate the interaction in natural language, we implemented a dialogue manager and a natural language understanding module, using the IBM Watson platform because it supports all the required services. To detail the proposed solution, we have presented an application example that describes how the dialogue manager would be integrated with the ITS. In the future, we plan to conduct a pilot study with 25 students of the first semester of the Forestry Engineering Degree of the Universidad Politécnica de Madrid. We will select a part of the practice that is carried out in the Virtual Biotechnology Laboratory, in which the dialogue manager will help students and answer their ques-tions. In this way we will evaluate the effectiveness and robustness of the proposed approach. The results of this study will serve to identify and address any inconvenience, so that we can successfully conduct a second study in which we will employ the dialogue system in the entire practice. To formulate the evaluation metrics, we plan to use the Goal, Question, Metric (GQM) methodology [25]. This methodology is used to identify which metrics are necessary to achieve the objectives of the pilot study. The evaluation of the first study will be conducted through an experiment to test the following research questions: a) Is the ITS robust? b) Does the ITS answer the students’ questions properly?. Dialogues will be collected from the ITS log file and analysed to evaluate the quality of the ITS answers. Later, after analysing the results of the first study, we will define the research questions for the second study.

ACKNOWLEDGEMENTS

Paladines would like to acknowledge financial support from the Universidad Estatal del Sur de Manabí.

REFERENCES

Clemente, J., Ramírez, J. and de Antonio, A. (2011) 'A proposal for student modeling based on ontologies and diagnosis rules', *Expert Systems with Applications*, 38(7), pp. 8066–8078.

Dalgarno, B. (2002) 'The potential of 3D virtual learning Environments: A constructivist analysis', *Electronic Journal of Instructional Science and Technology*, 5(2), pp. 1–19.

Dalgarno, B. and Lee, M. J. W. (2010) 'What are the learning affordances of 3-D virtual environments?', *British Journal of Educational Technology*, 41, pp. 10–32. doi: 10.1111/j.1467-8535.2009.01038.x.

Duncan, I., Miller, A. and Shangyi, J. (2012) 'A taxonomy of virtual worlds usage in education', *British Journal of Educational Technology*, 43(6), pp. 949–964.

Dzikovska, M. *et al.* (2014) 'BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic Electricity and Electronics', *Inter. J. Artif. Intell. Educ.*, 24(3), pp. 284–332.

Evers, M. and Nijholt, A. (2000) 'Jacob - An animated instruction agent in virtual reality', in Tan, T., Shi, Y., and Gao, W. (eds) *Advances in Multimodal Interfaces - ICMI 2000*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 526–533.

Gertner, A. S. and VanLehn, K. (2000) 'Andes: A coached problem solving environment for Physics', in Gauthier, G., Frasson, C., and VanLehn, K. (eds) *Proc. 5th Inter. Conf. Intelligent Tutoring System*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 133–142. doi: 10.1007/3-540-45108-0_17.

Glass, M. (2001) 'Processing language input in the CIRCSIM-Tutor intelligent tutoring system', in Al, J. D. M. et (ed.) *Proc. Artif. Intell. in Educ.* San Antonio: IOS Press, pp. 210–221.

Graesser, A. *et al.* (2005) 'AutoTutor: An Intelligent Tutoring System With Mixed-Initiative Dialogue', *IEEE Transactions on Education*, 48(4), pp. 612–618. doi: 10.1109/TE.2005.856149.

Graesser, A. C. *et al.* (2001) 'Intelligent tutoring systems with conversational dialogue', *AI Mag.*, pp. 39–51.

Gu, Y., Sosnovsky, S. and Ullrich, C. (2015) 'SafeChild: An intelligent virtual reality environment for training pedestrian safety skills', in Conole, G. et al. (eds) *Design for Teaching and Learning in a Networked World*. Cham: Springer International Publishing, pp. 141–154.

IBM (2016) *Natural Language Understanding*. Available at: https://www.ibm.com/watson/services/natural-language-understanding/ (Accessed: 15 September 2017).

IBM (2018) *Watson Assistant*. Available at: https://console.bluemix.net/docs/services/assistant/getting-started.html#getting-started (Accessed: 3 October 2018).

Koplowitz, R. *et al.* (2018) *The Forrester New Wave™: Conversational Computing Platforms, Q2 2018*.

Litman, D. J. and Silliman, S. (2004) 'ITSPOKE: An Intelligent Tutoring Spoken Dialogue System', in *Demonstration Papers at HLT-NAACL 2004*. Stroudsburg, PA, USA: Association for Computational Linguistics (HLT-NAACL--Demonstrations '04), pp. 5–8.

Los Arcos, J. L. *et al.* (2000) 'LAHYSTOTRAIN: Integration of virtual environments and ITS for surgery training', in Gauthier, G., Frasson, C., and VanLehn, K. (eds) *Int. Conf. Intelligent Tutoring System*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 43–52.

Mallios, S. and Bourbakis, N. (2016) 'A survey on human machine dialogue systems', in *Proc. 7th Inter. Conf. Inform. Intell. Syst. & Applicat.*, pp. 1–7.

Mitrovic, A. (2005) 'The Effect of Explaining on Learning: A Case Study with a Data Normalization Tutor', in *Int. Conf. on Artificial Intelligence in Education (AIED)*. Amsterdam, The Netherlands, The Netherlands: IOS Press, pp. 499–506.

Pioch, N., Roberts, B. and Zeltzer, D. (1997) 'A virtual environment for learning to pilot remotely operated vehicles', in *Proc. Int. Conf. Virtual Systems and MultiMedia*, pp. 218–226.

Rickel, J. *et al.* (2002) 'Collaborative discourse theory as a foundation for tutorial dialogue', in *Proc. 6th Int. Conf. on Intelligent Tutoring Systems*. Springer-Verlang, pp. 542–551.

Rickel, J. and Johnson, W. L. (1999) 'Animated agents for procedural training in virtual reality: Perception, cognition, and motor control', *Applied Artificial Intelligence*, 13, pp. 343–382.

Riofrío-Luzcando, D. (2012) *Diseño e implementación de un laboratorio virtual de Biotecnología*. M.S. thesis, ETSII, Univ. Politécnica de Madrid, Madrid, España.

Rosé, C. *et al.* (2001) 'Interactive conceptual tutoring in Atlas-Andes', in Morre, J. D., Redfield, C. L., and Johnson, W. L. (eds) *Inter. Conf. Artifi. Intell. in Educ.* Amsterdam: IOS Press, pp. 256–266.

Singh, B. (2017) *Chat Bots - Designing intents and entities for your NLP models*. Available at: https://blogs.msdn.microsoft.com/brijrajsingh/2017/01/29/chat-bots-designing-intents-and-entities-for-your-nlp-models/ (Accessed: 8 February 2017).

VanLehn, K. *et al.* (2002) 'The architecture of Why2-Atlas: A coach for qualitative Physics essay writing', in Cerri, S. A., Gouardères, G., and Paraguaçu, F. (eds) *Inter. Conf. on Intelligent Tutoring Systems*. Springer Berlin Heidelberg, pp. 158–167.