



**HAL**  
open science

# A Design Proposition for Interactive Virtual Tutors in an Informed Environment

Joanna Taoum, Bilal Nakhal, Elisabetta Bevacqua, Ronan Querrec

► **To cite this version:**

Joanna Taoum, Bilal Nakhal, Elisabetta Bevacqua, Ronan Querrec. A Design Proposition for Interactive Virtual Tutors in an Informed Environment. 16th International Conference on Intelligent Virtual Agents (IVA 2016), Sep 2016, Los Angeles, CA, United States. pp.341-350, 10.1007/978-3-319-47665-0\_30 . hal-01386412

**HAL Id: hal-01386412**

**<https://hal.science/hal-01386412v1>**

Submitted on 29 Nov 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Design Proposition for Interactive Virtual Tutors in an Informed Environment

Joanna Taoum, Bilal Nakhal, Elisabetta Bevacqua, and Ronan Querrec

Université Bretagne Loire (UBL),  
Laboratoire en Sciences et Techniques de l'Information, de la Communication et de la  
Connaissance (Lab-STICC),  
Ecole Nationale d'Ingénieurs de Brest (ENIB),  
Centre Européen de Réalité Virtuelle (CERV), 25 Rue Claude Chappe, Plouzané  
France  
{taoum,nakhal,bevacqua,querrec}@enib.fr  
<http://www.cerv.fr/>

**Abstract.** This paper introduces a new research work that aims to improve embodied conversational agents with tutor behavior by endowing them with the capability to generate feedback in pedagogical interactions with learners. The virtual agent feedback and the interpretation of the user's feedback are based on the knowledge of the environment (informed virtual environment), the interaction and the pedagogical strategies structured around classical intelligent tutoring system models. We present our first steps to implement our proposed architecture based on a model of informed virtual environment. We also describe the ideas that will guide the design of the Tutor Behavior. The planned evaluation method and a first application are also presented.

**Keywords:** Embodied Conversational Agents; Intelligent Tutoring System; Feedback; Informed Virtual Environment

## 1 Introduction

In the last years, virtual reality (VR) has acquired more and more interest as one of the most potential propositions to change and improve education. Previous research works have shown that this new technology seems to have a positive influence on learning in educational applications [1]. Moreover, the presence of interactive virtual agents also called Embodied Conversational Agents (ECA), by taking the role of tutors [2], seems to have positive effects on the student engagement [3] and the effectiveness of teaching [4].

In this work we aim to improve virtual tutor behavior by endowing it with the capability to generate and interpret feedback signals in pedagogical interactions with learners.

In human-human interactions, people emit regularly feedback signals in order to exchange information about the on-going communication [5]. For example, through feedback two interactants can inform each other about their understanding or their attitudinal reactions about the communicated content. This type of

behavior has been proven to be fundamental also in human-virtual agent interactions. For instance, previous researchers have shown that, to assure effective and satisfying interactions, a virtual agent must be able not only to speak but also to provide feedback signals while listening [6].

Previous research studies proposed models to automatically generate the interlocutor’s behavior during the interaction with the user. Most of these works focused on a subset of feedback signals called “backchannels”, which are multimodal signals provided by the listener without attempting to take the floor [7]. Particularly, previous works tried to determine the right moment to perform a backchannel signal, on the one hand according to the acoustic and visual signals shown by the speaker [8,9] and on the other hand according to the the content of the speaker’s speech [10]. In the first case the agent performs *reactive* backchannels deriving from perception processing, while in the second case the virtual character performs *responses* backchannels which consist in a more aware behavior generated by reasoning processing. In this work we want to improve the agent behavior by performing a wider set of feedback signals, such as multimodal signals (including backchannels) and short sentences which will make the agent take the turn. Moreover, we want these signals to be deliberate, based on a more reasoning process which takes into account not only the content of what the user can say but also the pedagogical model and the domain model of the environment. The choice of the feedback performed by the virtual tutor will be based on an inference of the the learner’s cognitive state and can be then consider as cognitive feedback [11].

For example, the agent could frown to show that the action that the learner is currently conducting is wrong. Thereby, the agent would show the learner that they are making a mistake and, at the same time, that it is attentive and watchful to their presence and actions. Another example, if the agent notices that the learner is not looking in the right direction, for instance where the object of interest is, it could point to the object to guide the learner.

This work aims to define a model for virtual tutor able, on the one hand, to observe the learner’s actions in order to provide multimodal feedback, and, on the other hand, to recognize the learners’ feedback in order to estimate their level of understanding (related to cognitive state). The interaction between the tutor and the learner takes place in an intelligent virtual environment [12]. Unlike traditional virtual environments that are represented only as a set of 3D elements, intelligent virtual environments provide semantic information. Classically this semantic covers the way the user can interact with the object [13]. The goal of intelligent virtual environment is to provide agents with high level of information, which permits them to have more intelligent behaviors. As we consider that the environment can be defined independently of the agent’s behaviors, and that the intelligent aspect belongs to agent’s behaviors, based on the semantic information in the environment, in the rest of this article we will use the term of informed virtual environment instead of intelligent virtual environment.

In the next section we present the model we use to represent the informed virtual environment. Then in section 3 we describe our proposed architecture

highlighting the work already done and ongoing. In the fourth and fifth sections, we describe our planned evaluation method and the first integrated application applied in the domain of education.

## 2 Informed Virtual Environment Model

In this work we choose MASCARET [14], Multi-Agent System for Collaborative, Adaptive & Realistic Environments for Training, to define the informed (intelligent) virtual environment. We use it as a knowledge base for agents and as an agent architecture.

MASCARET permits to design the semantic of virtual environments (VE). It is a virtual reality meta-model based on the Unified Modeling Language (UML) meta-model. MASCARET covers all the aspects of VEs semantic representation: ontology of the domain, structure of the environment, behavior of the entities, both users and agents interactions and activities. These aspects represents the domain model. In MASCARET we consider pedagogy as a specific domain model. We use then the same language (UML) to describe the domain and the pedagogical model. The pedagogical model is represented in our work by the pedagogical scenario. Koper [15] considers that a pedagogical scenario is composed of five main elements: pedagogical objectives, pedagogical prerequisites, pedagogical activities, pedagogical organizations and pedagogical environments. In MASCARET pedagogical scenarios are implemented through a chain of actions and activities. Those actions and activities can be either pedagogical actions, like explaining a resource, or domain actions, like manipulating an object.

In MASCARET, class diagrams are used to describe the different types of entities, their properties and the structure of the environment. Asynchronous discreet entity behaviors are defined through state machines. Activities are designed as predefined collaborative scenarios (called procedures), which represent plans of action for virtual agents or instructions provided to users for assisting them. The way the activity is interpreted by the agents is defined using specific agent behaviors. It is important to notice that in MASCARET, any entity which acts on the environment is called an agent.

MASCARET is a UML profile (extension) for virtual environments. The domain model is defined using a classical UML modeler and exported in the XMI normalized file format. Classically, to define a VE, computer scientists design the scene and all behaviors that occur in the VE. By using MASCARET, end-users (pedagogue, domain expert and domain trainer) are directly involved in the creation of the VE. The work flow to design a virtual environment for learning is presented in figure 1.

The pedagogue defines the pedagogical actions that can be used to guide or correct the trainee in the VE, as well as the pedagogical action forms (typical sequences of actions, reactions and interactions with the objects of the system). These actions are independent from the application domain, from the technological environment and from the pedagogical strategies. However, they depend on the type of learning environment, for example interactive simulations.

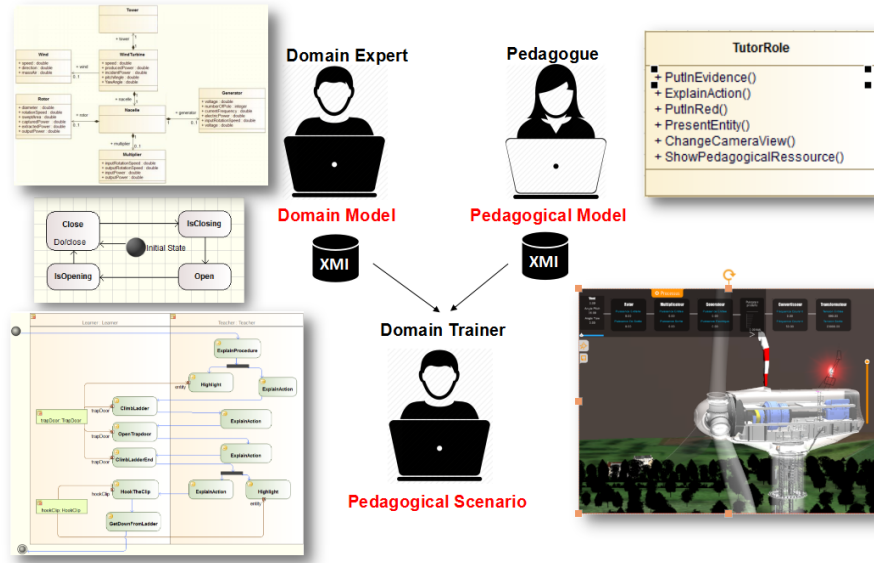


Fig. 1. MASCARET work flow to design a virtual environment for learning.

The domain expert, who knows the activity which has to be learned, formalizes the sequence of actions and interactions with the objects of the environment. They also describe good practices and procedures that have to be learned and different behaviors (proactive or reactive) of the objects. This description is independent from the execution platform.

The domain trainer defines pedagogical scenarios (the sequence of situations in which the trainee acts in the environment) and the pedagogical assistance provided by the system in real time. To define the scenarios, the domain trainer uses (1) the environment and the objects it contains, (2) the potential actions of the learner on the objects and the good practices (defined by the domain expert), and (3) the generic pedagogical actions (defined by the pedagogue).

### 3 Architecture

In this work we aim to add new features to MASCARET in order to integrate a virtual embodied tutor. This tutor has to be able to apply pedagogical strategies and to perform appropriate verbal and non-verbal behaviors while speaking, listening and observing the user. In particular, we want this tutor to be able to provide feedback based on the knowledge of the environment and the interaction. Figure 2, shows the class diagram of the global architecture in which we are developing our work (in blue the elements already integrated and in green what we are working on at the moment).

All entities which can act on the environment are instances of the **Agent** class (we call them simply *agents*). Agents have knowledge about the environment.

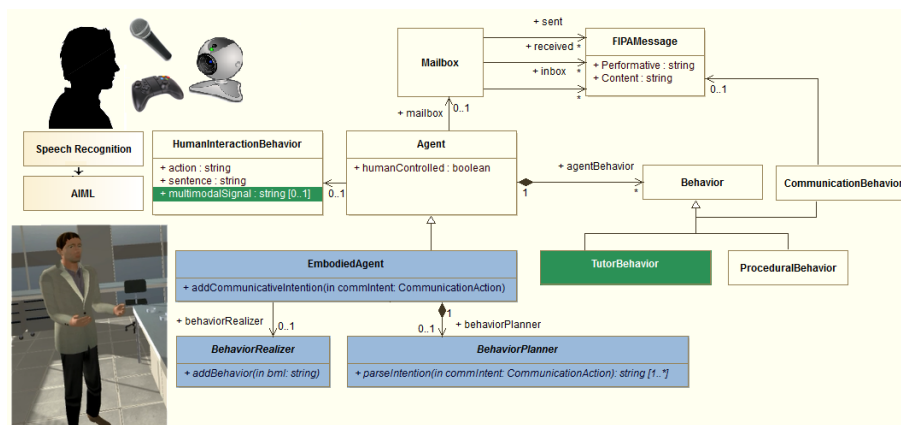


Fig. 2. Proposed global architecture

The structure of this knowledge uses the semantic of the domain and pedagogical model explained in section 1. Agents have behaviors (**Behavior**). They have at least a communication behavior (**CommunicationBehavior**) and a procedural behavior (**ProceduralBehavior**), but MASCARET permits to add other generic behaviors (for example a tutor behavior). The communication behavior allows the agents to exchange messages through a **Mailbox**. Agents can communicate using FIPA messages and FIPA-SL content language<sup>1</sup>. FIPA is a formal and normalized language for agent communication. For example, an agent asking for the value of a property of an entity, sends this FIPA message:

"QueryRef: ((iota ?propertyName (slot ?propertyName ?entity)))",

where *iota* and *slot* are keywords. The procedural behavior permits the agent to realize a procedure. By using MASCARET, this procedure is considered as an explicit knowledge during its execution. Agents are then able to execute a procedure and they can also reason about it.

Two types of agents can be instantiated: agents representing human users and embodied agents. Human users are represented as an agent which is not autonomous (**ControlByHuman = true**). They speak in natural language and for such a reason, each agent that represents a user, is provided with an interface (**HumanInteractionBehavior**) which can receive the sentences uttered by the user and translate them to FIPA-SL content by using Artificial Intelligence Markup Language (AIML)<sup>2</sup>. At present, the **HumanInteractionBehavior** interface is connected to RealSense by Intel®, which can recognize also some user's non-verbal signals (such as facial expressions, head orientation, etc.). This interface can also collect the user's actions on the virtual environment, performed, for instance, through a mouse or any other VR peripheral.

<sup>1</sup> <http://www.fipa.org/>

<sup>2</sup> <http://www.alicebot.org/>

Embodied agents can be instantiated in the VE through a humanoid body (**EmbodiedAgent**) or without a physical representation (**Agent**). When embodied agents communicate with the user, they cannot use FIPA messages, they have to use natural language. For such a reason, following the SAIBA architecture [16], each embodied agent is provided with a **BehaviorPlanner** and a **BehaviorRealizer**. The former receives the tutor communicative intentions and generates the verbal and non-verbal signals needed to transmit them. The latter realizes these signals through the tutor body. The **BehaviorPlanner** and **BehaviorRealizer** classes are abstract, which means that we can propose several implementations according to the ECA platform that we want to use (Greta [17], Virtual Human Toolkit [18], MARC [19], etc). At present, an implementation of these classes to use the virtual ECA Greta has been successfully integrated.

Our main contribution to this architecture will consist in defining a tutor behavior. This behavior will use the communication and procedural behavior. However it will be able to enrich the execution of pedagogical scenario, by generating cognitive feedback or to interrupt its execution in order to realize other pedagogical actions based on the learner's feedback. The main ideas of this behavior are describe in the following section.

### 3.1 Tutor Behavior

Our proposed tutor behavior relies on classical intelligent tutoring system (ITS) models [20] and on an inference of the cognitive state of the user while learning in a VE.

Classical ITS are composed of the domain model, the pedagogical model and the learner model. The domain model is represented by the informed virtual environment. It is the knowledge base of the agent that provides it with knowledge about the domain procedures, the actions and the entities states. The pedagogical model is represented in our work by the pedagogical scenario and the set of generic pedagogical actions. The main elements that define the learner model are related to their curriculum, the history of actions realized during the current exercise and the learner profile. In our case (procedural learning on technical systems) the best way to learn a procedure is to repeat several times the execution of the procedure. That is why the curriculum stocks the number of repetitions done by the learner. The curriculum stocks also all the exercises (learned procedures) in a Learning Management System (LMS). This permits to identify the level of expertise of the learner. Our major contribution to the learner model, will be to add the learner feedback. Fitts [21] and Anderson [22] proposed to structure the cognitive processes involved in learning in three stages. The inference of the cognitive state of the student by the tutor is based on these learning stages. For example, during the first stage (cognitive stage), the main cognitive process is based on the identification of objects to manipulate. At the second stage (associative stage) the learner organizes the information to be processed (i.e. list of actions to memorize) using a logical structure (i.e. hierarchy and action's goal) to store it entirely in working memory. During the last stage (autonomous stage) transformation of declarative knowledge to procedural knowledge happens.

The objective of the tutor’s behavior is to deduce, according to learner’s feedback, in which stage the learner is and which exact cognitive process occurs. The result of this inference is stored in the learner model. The tutor behavior is defined according to whom (tutor or learner) takes the initiative to interact. Most of the time, the tutor agent can be the initiator of the interaction. For example, the tutor can start by greeting the learner and then it can give them some general information and objectives about the procedure that the learner has to realize. This pedagogical scenario, which is predefined by a real teacher, is executed by the procedural behavior. The tutor behavior checks the user feedback, each time the tutor agent realizes an action.

For example, if the learner frowns after the tutor explanation of the next action, according to this feedback and to the learner model, the agent will be able to decide to realize another pedagogical action. In case of novice learners, the agent will automatically provide more information about the object to manipulate. Whereas, in case of expert learners, the tutor will focus on explaining the goal of the sub-part of the procedure to learn. Furthermore, when the tutor agent asks the learner to manipulate an object, if the learner frowns (or expresses a negative feedback), the tutor can decide to provide more information, for example it can explain the role of the object. At every moment the tutor behavior can be affected by an interruption from the learner. The learner can either interact with the environment or the tutor, by realizing a domain action. After every domain action done by the learner, the tutor checks if the action performed and the object manipulated are correct. Depending on the outcome of this check and on the learner model, the tutor can choose to perform a feedback. For example, when the tutor agent knows that the next action the learner has to realize is to manipulate the object O1, but instead the learner manipulates the object O2, the tutor agent can show a negative feedback, like saying “Wrong”, or simply frowning.

## 4 Evaluation Method

Like some other previous evaluation works [23], based on objective performance measures, we aim to validate objectively the impact of our proposition on the performance of the learner. For that we envisage to apply the experimental protocols defined by Hoareau et al. [24]. In this study, the researchers evaluate the interest of virtual reality for learning procedures on the base of objective performance measures, like the execution time, consulting assistance, the number of errors. The hypotheses that Hoareau et al. evaluated rely on the learning stages presented in section 3.1. Following this evaluation, learning in virtual reality has been validated and even the transfer of the acquired procedure in a virtual environment to a real situation. However, Hoareau et al.’s evaluations and experiments were conducted using non immersive virtual reality devices, like PCs with low graphics quality. In our research, we are planning to do experiments with good graphics quality, and different immersion levels using PCs, virtual



reality head mounted display and CAVE<sup>3</sup>. After identifying the best situation for learning, we plan to integrate an embodied virtual agent with the tutor behavior explained in section 3.1 in order to evaluate the influence of the presence of the virtual agent when learning a procedure.

## 5 Application



**Fig. 3.** Screen-shots from the blood analysis procedure

A first application was developed and will integrate the behavior described in the section 3. It is an informed virtual environment for procedural learning for blood analysis. The procedures to learn involve actions on the automaton and the reagents preparation. Figure 3 contains two screen-shots taken from this application. In those two pictures the virtual tutor (from the Greta platform) is placed in the middle of the scene. The procedure involves actions applied on the automaton that is positioned at its left (like turning on the machine, opening the drawer, etc.) and actions to prepare the reagents that are at the agent's right, on the bench (like vials, test tubes, etc.). In the left picture, the virtual tutor informs about the first action to perform. In the picture on the right, the learner asks for more information about the object to manipulate (Neoplastine). Through its communication behavior, the virtual agent interprets this question and, using its knowledge about the environment, describes the object.

## 6 Conclusion

In this paper we presented an on-going work to endow a virtual agent tutor with the capability to provide and interpret feedback in a pedagogical interaction

<sup>3</sup> Cave Automatic Virtual Environment

with learners. We showed our first steps to implement the proposed architecture based on a model of informed virtual environment. We also described the ideas that will guide the design of the tutor behavior and its evaluation.

### Acknowledgments

This work is supported by a grant from the Région Bretagne and by the ANR, Agence Nationale de la Recherche (INGREDIBLE project).

### References

1. Limniou, M., Roberts, D., Papadopoulos, N.: Full immersive virtual environment CAVE[TM] in chemistry education. *Computers & Education* **51**(2) (2008) 584–593
2. Johnson, W., Rizzo, P., Bosma, W., Kole, S., Ghijsen, M., van Welbergen, H.: Generating socially appropriate tutorial dialog. In: *ISCA Workshop on Affective Dialogue Systems*, Berlin, Heidelberg (2004) 254–264
3. Rowe, J., McQuiggan, S., Mott, B., Lester, J.: Motivation in narrative-centered learning environments. *Proceedings of the workshop on narrative learning environments*, AIED (2007) 40–49
4. Kokane, A., Singhal, H., Mukherjee, S., Reddy, G.: Effective e-learning using 3D virtual tutors and webRTC based multimedia chat. In: *International Conference on Recent Trends in Information Technology (ICRTIT)*. (2014) 1–6
5. Allwood, J., Nivre, J., Ahlsén, E.: On the semantics and pragmatics of linguistic feedback. *Journal of Semantics* **9**(1) (1993) 1–30
6. Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S., Morales, M., van der Werf, R., Morency, L.: Virtual rapport. In: *6th International Conference IVA*. Volume 4133 of *LNAI*. (2006) 14–27
7. Yngve, V.: On getting a word in edgewise. *Papers from the Sixth Regional Meeting of the Chicago Linguistic Society* (1970) 567–577
8. Bevacqua, E., de Sevin, E., Hyniewska, S., Pelachaud, C.: A listener model: introducing personality traits. *Journal of Multimodal User Interfaces*, special issue *Interacting ECAs* **6**(1) (2012) 27–38
9. Morency, L.P., de Kook, I., Gratch, J.: Predicting listener backchannels: A probabilistic multimodal approach. In: *8th International Conference IVA*. Volume 5208 of *LNCS*. (2008) 176–190
10. Kopp, S., Allwood, J., Grammer, K., Ahlsen, E., Stocksmeier, T.: Modeling embodied feedback with virtual humans. In: *Modeling communication with robots and virtual humans*. (2008) 18–37
11. Doherty, M.E., Balzer, W.K.: Cognitive feedback. *Advances in psychology*, 54. *Human judgment: The SJT view* (1988) 163–197
12. Aylett, R., Cavazza, M.: Intelligent virtual environments - a state-of-the-art report. In Duke, D., Scopigno, R., eds.: *Proceedings of Eurographics '01*. (2001)
13. Kallmann, M.: *Object Interaction in Real-Time virtual Environments*. PhD Thesis, Swiss Federal Institute of Technology EPFL, Lausanne, Switzerland (2001)
14. Chevaillier, P., Trinh, T., Barange, M., Devillers, F., Soler, J., De Loor, P., Querrec, R.: Semantic modelling of virtual environments using MASCARET. In: *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems*, IEEE VR, Singapore. (2001)

15. Koper, R., van Es, R.: Modelling units of learning from a pedagogical perspective. *Online Education Using Learning Objects* **40** (2004) 40–52
16. Kopp, S., Krenn, B., Marsella, S., Marshall, A.N., Pelachaud, C., Pirker, H., Thórisson, K., Vilhjálmsson, H.: Towards a common framework for multimodal generation: The behavior markup language. In: *Proceedings of IVA*. Volume 4133 of LNCS., Springer, Berlin Heidelberg (2006) 205–217
17. Niewiadomski, R., Bevacqua, E., Mancini, M., Pelachaud, C.: Greta: an interactive expressive eca system. In: *8th International Conference AAMAS*. (2009) 1399–1400
18. Gratch, J., Hartholt, A., Dehghani, M., Marsella, S.C.: *Virtual Humans: A New Toolkit for Cognitive Science Research*. In: *Cognitive Science*. (2013)
19. Courgeon, M.: *Marc: computational models of emotions and their facial expressions for real-time affective human-computer interaction*. PhD Thesis, Université Paris Sud - Paris XI (2011)
20. Wenger, E.: *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos, California (1987)
21. Fitts, P.: *Categories of Human Learning*. Academic Press, New York (1964)
22. Anderson, J.: *The Architecture Of Cognition*. Harvard University Press (1983)
23. Atkinson, R.K.: Optimizing learning from examples using animated pedagogical agents. *Journal of Educational Psychology* **94**(2) (2002) 416–427
24. Hoareau, C., Ganier, F., Querrec, R., Corre, F.L., Buche, C.: Evolution of cognitive load when learning a procedure in a virtual environment for training. *6th International Cognitive Load Theory Conference (ICLTC'13)* (2013)