



HAL
open science

Une proposition de conception pour les tuteurs virtuels interactifs dans un environnement informé

Joanna Taoum, Bilal Nakhal, Elisabetta Bevacqua, Ronan Querrec

► To cite this version:

Joanna Taoum, Bilal Nakhal, Elisabetta Bevacqua, Ronan Querrec. Une proposition de conception pour les tuteurs virtuels interactifs dans un environnement informé. 11e Journées de l'Association Française de Réalité Virtuelle (AFRV 2016), Oct 2016, Brest, France. hal-01413214

HAL Id: hal-01413214

<https://hal.science/hal-01413214>

Submitted on 9 Dec 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.

Une proposition de conception pour les tuteurs virtuels interactifs dans un environnement informé

Joanna Taoum*
Lab-STICC, ENIB, UBL

Bilal Nakhal†
Lab-STICC, ENIB, UBL

Elisabetta Bevacqua‡
Lab-STICC, ENIB, UBL

Ronan Querrec§
Lab-STICC, ENIB, UBL

ABSTRACT

Cet article présente un travail de recherche qui vise à améliorer les agents conversationnels animés ayant un comportement de tuteur, en les munissant de la capacité à générer des rétroactions, en anglais *feedbacks*, lors d'interactions pédagogiques avec des apprenants. Les *feedback* de l'agent virtuel et l'interprétation des *feedbacks* de l'utilisateur sont basés sur la connaissance de l'environnement (environnement virtuel informé), l'interaction et les stratégies pédagogiques, qui sont structurées autour des modèles classiques de tuteur intelligent (ITS). Nous présentons nos premières étapes de mise en œuvre de notre architecture qui se base sur un modèle d'environnement virtuel informé. Nous décrivons également les idées qui guideront la conception du comportement du tuteur. La méthode d'évaluation prévue ainsi qu'une première application sont également présentées.

Keywords: Agents Conversationnels Animés; Tuteur Intelligent; Feedback ; Environnement Virtuel Informé

1 INTRODUCTION

Depuis ces dernières années, la réalité virtuelle (VR) est une des technologies qui a le plus de potentiel à changer et améliorer la formation. Des précédents travaux de recherche ont montré que cette nouvelle technologie semble avoir une influence positive sur l'apprentissage [19]. En outre, la présence des agents virtuels interactifs également appelés Agents Conversationnels Animés (ACA), en prenant le rôle de tuteur [13], semble avoir des effets positifs sur l'engagement des étudiants [22] et sur l'efficacité de l'enseignement [15].

Dans ce travail, nous visons à améliorer le comportement de tuteur virtuel en lui donnant la capacité de générer des *feedbacks* et d'interpréter ceux de l'utilisateur lors des interactions pédagogiques avec les apprenants.

Dans les interactions humain-humain, les participants émettent régulièrement des signaux de *feedbacks* pour échanger des informations sur la communication en cours [1]. Par exemple, grâce aux *feedbacks*, deux interlocuteurs peuvent s'informer mutuellement de leur compréhension ou de leurs réactions sur le contenu communiqué. Il a été démontré que ce type de comportement est également fondamental dans les interactions entre humains et agents virtuels. Par exemple, Gratch et al. [11] ont montré que, pour assurer des interactions efficaces et satisfaisantes, un agent virtuel doit être en mesure non seulement de parler, mais aussi de fournir des signaux de *feedbacks* en écoutant.

Des études antérieures ont proposé des modèles pour générer automatiquement le comportement de l'interlocuteur lors de l'interaction avec l'utilisateur. La plupart de ces travaux ont porté

sur un sous-ensemble de signaux de *feedbacks* appelés *backchannels*, qui sont des signaux multimodaux fournis par l'auditeur sans essayer de prendre la parole [24]. En particulier, les travaux précédents ont tenté de déterminer le bon moment pour émettre un signal de *backchannel*, d'une part en fonction des signaux acoustiques et visuels présentés par l'orateur [5, 20] et d'autre part en fonction du contenu du discours de l'orateur [17]. Dans le premier cas, l'agent effectue des *backchannels* réactifs provenant de l'interprétation de la perception, alors que dans le second cas, le personnage virtuel effectue des *responses backchannels* qui consistent en un comportement plus conscient généré par un raisonnement. Dans ce travail, nous voulons améliorer le comportement de l'agent en effectuant un ensemble plus large de *feedback*, tels que les signaux multimodaux (y compris les *backchannels*) et des phrases courtes qui conduiront l'agent à prendre le tour de parole. De plus, nous voulons que ces signaux soient délibérés, sur la base d'un processus de raisonnement qui prend en compte non seulement le contenu de ce que l'utilisateur peut dire, mais aussi le modèle pédagogique et le modèle du domaine de l'environnement. Le choix des *feedbacks* effectué par le tuteur virtuel sera basé sur une inférence de l'état cognitif de l'apprenant et peut être alors considéré comme un *feedback* cognitif [8].

Par exemple, l'agent pourrait froncer les sourcils pour montrer que l'action que l'apprenant mène actuellement est erronée. De ce fait, l'agent montrerait à l'apprenant qu'il est en train de faire une erreur et, en même temps, qu'il est attentif et vigilant à sa présence et ses actions. Un autre exemple, si l'agent constate que l'apprenant ne regarde pas dans la bonne direction, par exemple, là où l'objet d'intérêt se trouve, il pourrait pointer vers l'objet pour guider l'apprenant.

Ce travail vise à définir un modèle de tuteur virtuel capable, d'une part, d'observer les actions de l'apprenant afin de fournir des *feedbacks* multimodaux, et d'autre part, de reconnaître les *feedbacks* des apprenants afin d'estimer leur niveau de compréhension (lié à l'état cognitif). L'interaction entre le tuteur et l'apprenant se déroule dans un environnement virtuel intelligent [4]. Contrairement aux environnements virtuels traditionnels, qui sont représentés comme un ensemble d'éléments 3D, les environnements virtuels intelligents fournissent des informations sémantiques. Classiquement ces sémantiques couvrent la façon dont l'utilisateur peut interagir avec un objet [14]. Le but de l'environnement virtuel intelligent est de fournir aux agents un haut niveau d'information leur permettant d'avoir des comportements plus intelligents. Comme nous considérons que l'environnement peut être défini indépendamment des comportements de l'agent, et que l'aspect intelligent appartient aux comportements des agents, s'appuyant sur la sémantique des informations dans l'environnement, dans le reste de cet article, nous utiliserons le terme d'environnement virtuel informé au lieu d'environnement virtuel intelligent.

Dans la section suivante, nous présentons le modèle que nous utilisons pour représenter l'environnement virtuel informé. Ensuite, dans la section 3, nous décrivons l'architecture proposée, soulignant le travail déjà accompli et celui en cours. Dans la quatrième et cinquième section, nous décrivons la méthode d'évaluation prévue et la première application appliquée dans le domaine de l'éducation.

*e-mail: taoum@enib.fr

†e-mail: nakhal@enib.fr

‡e-mail: bevacqua@enib.fr

§e-mail: querrec@enib.fr

2 MODÈLE D'ENVIRONNEMENT VIRTUEL INFORMÉ

Dans ce travail, nous avons choisi d'utiliser MASCARET [6], Multi-Agent System for Collaborative, Adaptive & Realistic Environments for Training, pour définir l'environnement virtuel informé (intelligent). Nous l'utilisons comme une base de connaissances pour les agents et comme une architecture d'agent.

MASCARET permet de concevoir la sémantique des environnements virtuels. C'est un méta-modèle de réalité virtuelle basé sur le méta-modèle UML (*Unified Modeling Language*). MASCARET couvre tous les aspects de la représentation sémantique de l'environnement virtuel : l'ontologie du domaine, la structure de l'environnement, le comportement des entités, et à la fois les interactions et les activités de l'utilisateur et des agents. Ces aspects représentent le modèle du domaine. Dans MASCARET nous considérons la pédagogie comme un modèle du domaine spécifique. Nous utilisons ensuite le même langage (UML) pour décrire le modèle du domaine et le modèle pédagogique. Le modèle pédagogique est représenté dans notre travail par le scénario pédagogique. Koper [16] considère qu'un scénario pédagogique est composé de cinq éléments principaux : objectifs pédagogiques, prérequis pédagogiques, activités pédagogiques, organisations pédagogiques et environnements pédagogiques. Dans MASCARET les scénarios pédagogiques sont implémentés par une chaîne d'actions et d'activités. Ces actions et activités peuvent être soit des actions pédagogiques, comme expliquer une ressource, ou des actions de domaine, comme manipuler un objet.

Dans MASCARET, les diagrammes de classes sont utilisés pour décrire les différents types d'entités, leurs propriétés et la structure de l'environnement. Les comportements discrets et asynchrones des entités sont définies par des machines à état. Les activités sont conçues comme des scénarios de collaboration prédéfinis (appelés procédures), qui représentent des plans d'action pour les agents virtuels ou des instructions fournies aux utilisateurs pour les aider. La façon dont l'activité est interprétée par les agents est définie en utilisant des comportements des agents spécifiques. Il est important de noter que, dans MASCARET, toute entité qui agit sur l'environnement est considéré comme un agent.

MASCARET est un profil UML (extension) pour les environnements virtuels. Le modèle du domaine est défini à l'aide d'un modèleur UML classique et exporté dans le format de fichier normalisé XML. Classiquement, pour définir un environnement virtuel, les informaticiens conçoivent la scène et tous les comportements qui se produisent dans l'environnement virtuel. En utilisant MASCARET, les utilisateurs finaux (pédagogue, expert du domaine et formateur du domaine) sont directement impliqués dans la création de l'environnement virtuel. Le flux de travail pour concevoir un environnement virtuel pour l'apprentissage est présenté dans la figure 1.

Le pédagogue est celui qui définit les actions pédagogiques qui peuvent être utilisés pour guider ou corriger l'apprenant dans l'environnement virtuel, ainsi que les formes d'action pédagogiques (séquences typiques des actions, des réactions et des interactions avec les objets du système). Ces actions sont indépendantes du domaine d'application, de l'environnement technologique et des stratégies pédagogiques. Cependant, elles dépendent du type d'environnement d'apprentissage, par exemple des simulations interactives.

L'expert du domaine est celui qui connaît l'activité qui doit être apprise, qui formalise la séquence d'actions et d'interactions avec les objets de l'environnement. Il décrit également les bonnes pratiques et les procédures qui doivent être apprises et les différents comportements (proactifs ou réactifs) des objets. Cette description est indépendante de la plate-forme d'exécution.

Le formateur est celui qui définit les scénarios pédagogiques (la séquence des situations dans lesquelles l'apprenant agit dans l'environnement) et l'aide pédagogique fournie par le système en

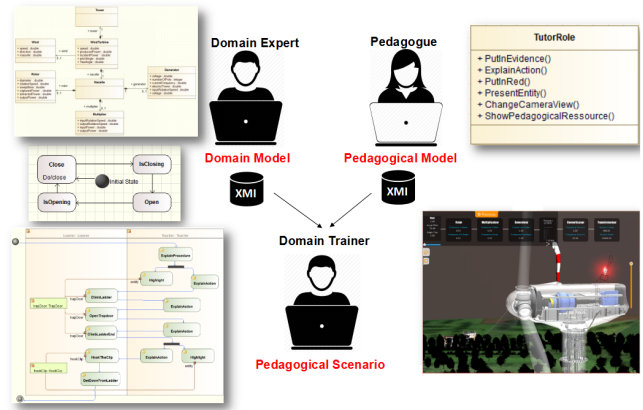


Figure 1: Flux de la conception d'un environnement virtuel pour l'apprentissage.

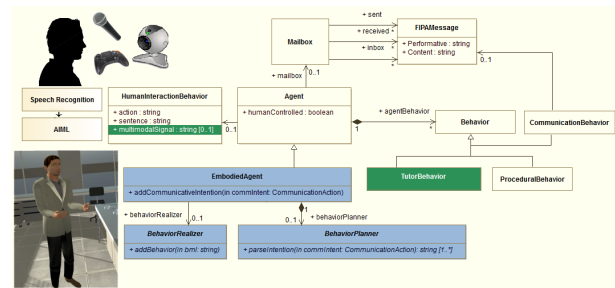


Figure 2: Architecture globale proposée

temps réel. Pour définir les scénarios, le formateur utilise (1) l'environnement et les objets qu'il contient, (2) les actions potentielles de l'apprenant sur les objets et les bonnes pratiques (définies par l'expert du domaine), et (3) les actions pédagogiques génériques (définies par le pédagogue).

3 ARCHITECTURE

Dans ce travail, nous visons à ajouter de nouvelles fonctionnalités à MASCARET, afin d'intégrer un tuteur virtuel incarné. Ce tuteur doit être capable d'appliquer des stratégies pédagogiques et d'effectuer des comportements verbaux et non verbaux appropriés, tout en parlant, en écoutant et en observant l'utilisateur. En particulier, nous voulons que ce tuteur soit capable de fournir des *feedbacks* en se basant sur la connaissance de l'environnement et de l'interaction. La figure 2 montre le diagramme de classe de l'architecture globale dans laquelle nous développons notre travail (en bleu les éléments déjà intégrés et en vert ceux sur lesquels nous travaillons en ce moment).

Toutes les entités qui peuvent agir dans l'environnement sont des instances de la classe Agent (nous les appelons simplement *agents*). Les agents ont des connaissances sur l'environnement. La structure de cette connaissance utilise la sémantique du domaine et le modèle pédagogique expliqué dans la section 1. Les agents ont des comportements (Behavior). Ils ont au moins un comportement de communication (CommunicationBehavior) et un comportement procédural (ProceduralBehavior), mais MASCARET permet d'ajouter d'autres comportements génériques (par exemple un comportement de tuteur). Le comportement de communication permet aux agents d'échanger des messages à travers un Mailbox. Les agents peuvent communiquer en utilisant

des messages FIPA et FIPA-SL ¹. FIPA est un langage formel et normalisé pour la communication entre agents. Par exemple, un agent demandant la valeur d'une propriété d'une entité, envoie ce FIPA message :

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

où `iota` et `slot` sont des mots clés.

Le comportement procédural permet à l'agent de réaliser une procédure. En utilisant MASCARET, cette procédure est considérée comme une connaissance explicite lors de son exécution. Les agents sont alors capables d'exécuter une procédure et ils peuvent aussi raisonner à ce sujet.

Deux types d'agents peuvent être instanciés : les agents représentant les utilisateurs humains et les agents incarnés. Les utilisateurs humains sont représentés comme un agent non-autonome (`ControlByHuman = true`). Ils parlent en langage naturel et pour cette raison, chaque agent représentant un utilisateur, est muni d'une interface (`HumanInteractionBehavior`) qui peut recevoir les phrases prononcées par l'utilisateur et les traduire en FIPA-SL, en utilisant l'Intelligence Artificielle Markup Language (AIML)². Actuellement, l'interface `HumanInteractionBehavior` est connectée à RealSense Intel®, qui peut reconnaître quelques signaux non verbaux des utilisateurs (tels que les expressions faciales, l'orientation de la tête, etc.). Cette interface peut également collecter les actions réalisées par l'utilisateur dans l'environnement virtuel, par exemple, par une souris ou tout autre périphérique de réalité virtuelle.

Les agents incarnés peuvent être instanciés dans l'environnement virtuel à travers un corps humanoïde (`EmbodiedAgent`) ou sans représentation physique (`Agent`). En communiquant avec l'utilisateur, les agents incarnés ne peuvent pas utiliser des messages FIPA, ils doivent utiliser le langage naturel. Pour cette raison, selon l'architecture Saiba [18], chaque agent incarné est fourni d'un `BehaviorPlanner` et d'un `BehaviorRealiser`. Le premier reçoit les intentions communicatives du tuteur et génère les signaux verbaux et non verbaux nécessaires pour les transmettre. Le dernier réalise ces signaux à travers le corps du tuteur. Les classes `BehaviorPlanner` et `BehaviorRealizer` sont abstraites, ceci signifie que nous pouvons proposer plusieurs implémentations en fonction de la plate-forme d'ACA que nous utilisons (GRETA [21], Virtual Human Toolkit [10], MARC [7], etc). Actuellement, des implémentations de ces classes pour utiliser les agents GRETA et Virtual Human Toolkit ont été intégrées (figure 3).

Notre contribution principale à cette architecture consistera à définir un comportement de tuteur. Ce comportement va utiliser le comportement de communication et le comportement procédural. Toutefois, l'agent sera capable d'enrichir l'exécution du scénario pédagogique, en générant des *feedbacks* cognitifs ou d'interrompre son exécution afin de réaliser d'autres actions pédagogiques en se basant sur les *feedbacks* de l'apprenant. Les idées principales de ce comportement sont décrit dans la section suivante.

3.1 Comportement du Tuteur

Le comportement du tuteur ici proposé se base sur les modèles classiques de tuteur intelligent (ITS) [23] et sur l'inférence de l'état cognitif de l'utilisateur lorsqu'il joue le rôle de l'apprenant dans un environnement virtuel.

Les ITS classiques sont composés du modèle du domaine, du modèle pédagogique et du modèle de l'apprenant. Le modèle du domaine est représenté par l'environnement virtuel informé. Il représente la base de connaissances de l'agent qui lui fournit des connaissances sur les procédures du domaine, les actions et les états des entités. Le modèle pédagogique est représenté



Figure 3: Scène intégrant deux ACAs (Virtual Human Toolkit et GRETA)

dans notre travail par des scénarios pédagogiques et l'ensemble des actions pédagogiques génériques. Les éléments principaux qui définissent le modèle de l'apprenant sont liés à sons cursus, l'historique des actions réalisées au cours d'un exercice et le profil de l'apprenant. Dans notre cas (apprentissage procédural sur système technique), la meilleure façon d'apprendre une procédure consiste à répéter plusieurs fois l'exécution de la procédure. C'est pour cela que le cursus sauvegarde le nombre de répétitions effectuées par l'apprenant. Le cursus sauvegarde également tous les exercices (procédures apprises) dans un système de gestion d'apprentissage (LMS). Cela permet d'identifier le niveau d'expertise de l'apprenant. Notre contribution majeure au modèle de l'apprenant, sera d'ajouter les *feedbacks* de l'apprenant. Fitts [9] et Anderson [2] ont proposé de structurer les processus cognitifs impliqués dans l'apprentissage en trois phases. L'inférence de l'état cognitif de l'étudiant par le tuteur est basé sur ces phases d'apprentissage. Par exemple, au cours de la première phase (phase cognitive), le processus cognitif principal est basé sur l'identification des objets à manipuler. À la deuxième phase (phase associative) l'apprenant organise l'information à traiter (par exemple la liste des actions à mémoriser) en utilisant une structure logique (ex. la hiérarchie et l'objectif de l'action) pour les stocker entièrement en mémoire de travail. La dernière phase (phase autonome) concerne la transformation des connaissances déclaratives en connaissance de la procédure.

L'objectif du comportement du tuteur est de déduire, selon les *feedbacks* de l'apprenant, dans quelle phase l'apprenant est et le processus cognitif exact qui se produit. Le résultat de cette inférence est stocké dans le modèle de l'apprenant. Le comportement du tuteur est défini selon celui qui (tuteur ou apprenant) prend l'initiative d'interaction. La plupart du temps, le tuteur peut être l'initiateur de l'interaction. Par exemple, le tuteur peut commencer par saluer l'apprenant et ainsi il peut lui donner des informations générales et les objectifs de la procédure que l'apprenant doit réaliser. Ce scénario pédagogique, qui est prédéfini par un vrai professeur, est exécuté par le comportement procédural du tuteur. A chaque fois que le tuteur réalise une action, le comportement du tuteur vérifie les *feedbacks* de l'utilisateur.

Par exemple, si l'apprenant fronce les sourcils après l'explication du tuteur sur l'action suivante, selon ces *feedbacks* et le modèle de l'apprenant, le tuteur peut décider de réaliser une autre action pédagogique. Dans le cas des apprenants novices, le tuteur fournit automatiquement plus d'informations sur l'objet à manipuler. Alors que, dans le cas des apprenants experts, le tuteur se concentrera sur l'explication de l'objectif de la sous-partie de la procédure à apprendre. En outre, lorsque le tuteur demande à l'apprenant de manipuler

¹<http://www.fipa.org/>

²<http://www.alicebot.org/>



Figure 4: Scène extraite de la procédure d'utilisation de l'automate.

un objet, si l'apprenant fronce les sourcils (ou exprime un *feedback* négative), le tuteur peut décider de fournir plus d'informations, par exemple, il peut expliquer le rôle de l'objet. A chaque instant le comportement du tuteur peut être affecté par une interruption de l'apprenant. L'apprenant peut interagir avec l'environnement ou avec le tuteur, en réalisant une action du domaine. Après chaque action du domaine faite par l'apprenant, le tuteur vérifie si l'action effectuée et l'objet manipulé sont corrects. Selon le résultat de cette vérification et selon le modèle de l'apprenant, le tuteur peut choisir d'émettre un *feedback*. Par exemple, lorsque le tuteur sait que la prochaine action que l'apprenant doit réaliser est de manipuler un objet O1, mais l'apprenant manipule l'objet O2, le tuteur peut montrer un *feedback* négative, par exemple il pourrait dire "c'est faux", ou simplement froncer les sourcils.

4 MÉTHODE D'ÉVALUATION

Comme certains précédents travaux d'évaluation [3] qui se sont basés sur des mesures de performance objective, nous visons à valider objectivement l'impact de notre proposition sur la performance de l'apprenant. Pour cela nous envisageons d'appliquer les protocoles expérimentaux définis par Hoareau et al. [12]. Dans cette étude, les chercheurs ont évalué l'intérêt de la réalité virtuelle pour les procédures d'apprentissage en se basant sur des mesures de performances objectives, comme par exemple le temps d'exécution, la consultation des assistances et le nombre d'erreurs. Les hypothèses que les chercheurs ont évaluées, se basent sur les phases d'apprentissage, présentées dans la section 3.1. Suite à cette évaluation, l'apprentissage en réalité virtuelle a été validé et de même le transfert de la procédure acquise dans un environnement virtuel à une situation réelle. Cependant, les évaluations et les expérimentations de Hoareau et al. ont été menées en utilisant des dispositifs de réalité virtuelle non-immersive, comme des PC avec une faible qualité graphique. Dans notre travail, nous envisageons de faire des expériences avec une bonne qualité graphique, et avec différents niveaux d'immersion en utilisant des PC, des casques de réalité virtuelle et un CAVE³. Après avoir identifié la meilleure situation pour l'apprentissage, nous prévoyons d'intégrer un agent virtuel incarné avec le comportement du tuteur expliqué à la section 3.1, afin d'évaluer l'influence de la présence de l'agent virtuel lors de l'apprentissage d'une procédure.

5 APPLICATION

Une première application a été développée et intégrera le comportement décrit dans la section 3. Il s'agit d'un environnement virtuel informé pour l'apprentissage procédural d'un automate pour l'analyse du sang. Les procédures à apprendre impliquent des actions sur l'automate et sur la préparation des réactifs. La figure 4 représente une capture d'écran extraite de cette application. Au milieu, un agent virtuel (issu de la plate-forme GRETA) guide

³Cave Automatic Virtual Environment



Figure 5: Scène extraite de la procédure d'ouverture de l'ancien pont tournant de Brest.

l'apprenant pour réaliser une procédure. Cette procédure fait intervenir des actions sur l'automate qui se trouve à la gauche de l'agent, (faire fonctionner la machine, ouvrir le tiroir, etc.) et des actions de préparation des réactifs qui sont positionnés à sa droite sur la pailasse (les flacons, les tubes à essai, etc.).

Dans l'image de gauche, le tuteur virtuel informe sur la première action à effectuer. Dans l'image de droite, l'apprenant demande plus d'informations sur l'objet à manipuler (Neoplastine). Grâce à son comportement de communication, l'agent virtuel interprète cette question et, en utilisant ses connaissances sur l'environnement, décrit l'objet.

6 CONCLUSION

Dans cet article, nous avons présenté un travail en cours pour doter un agent virtuel de la capacité de fournir des *feedbacks* et d'interpréter ceux émis par l'utilisateur dans une interaction pédagogique avec les apprenants. Nous avons montré nos premières étapes pour mettre en œuvre l'architecture proposée qui se base sur un modèle d'environnement virtuel informé. Nous avons également décrit les idées qui guideront la conception du comportement du tuteur et de son évaluation.

Dans le but de démontrer la généralité de notre modèle, nous envisageons également d'appliquer ce travail dans le domaine du patrimoine. Le modèle proposé permettra de réaliser des comportements d'agent ayant une connaissance non seulement sur l'apprentissage mais également sur un patrimoine socio-culturelle, comme par exemple le pont tournant de Brest. Les agents seront alors capable de participer aux activités et d'interagir d'une manière naturelle avec les visiteurs (figure 5).

ACKNOWLEDGEMENTS

Ce travail est financé par la Région Bretagne.

REFERENCES

- [1] J. Allwood, J. Nivre, and E. Ahlsén. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9(1):1–30, 1993.
- [2] J. Anderson. *The Architecture Of Cognition*. Harvard University Press, 1983.
- [3] R. K. Atkinson. Optimizing learning from examples using animated pedagogical agents. *Journal of Educational Psychology*, 94(2):416–427, 2002.
- [4] R. Aylett and M. Cavazza. Intelligent virtual environments - a state-of-the-art report. In D. Duke and R. Scopigno, editors, *In Proceedings of Eurographics '01*, 2001.

- [5] E. Bevacqua, E. de Sevin, S. Hyniewska, and C. Pelachaud. A listener model: introducing personality traits. *Journal of Multimodal User Interfaces, special issue Interacting ECAs*, 6(1):27–38, 2012.
- [6] P. Chevallier, T. Trinh, M. Barange, F. Devillers, J. Soler, P. De Loor, and R. Querrec. 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.
- [7] M. Courgeon. *Marc: computational models of emotions and their facial expressions for real-time affective human-computer interaction*. PhD Thesis, Université Paris Sud - Paris XI, 2011.
- [8] M. E. Doherty and W. K. Balzer. Cognitive feedback. *Advances in psychology*, 54. *Human judgment: The SJT view*, pages 163–197, 1988.
- [9] P. Fitts. *Categories of Human Learning*. Academic Press, New York, 1964.
- [10] J. Gratch, A. Hartholt, M. Dehghani, and S. C. Marsella. Virtual Humans: A New Toolkit for Cognitive Science Research. In *Cognitive Science*, 2013.
- [11] J. Gratch, A. Okhmatovskaia, F. Lamothe, S. Marsella, M. Morales, R. van der Werf, and L. Morency. Virtual rapport. In *6th International Conference IVA*, volume 4133 of *LNAI*, pages 14–27, 2006.
- [12] C. Hoareau, F. Ganier, R. Querrec, F. L. Corre, and C. Buche. Evolution of cognitive load when learning a procedure in a virtual environment for training. *6th International Cognitive Load Theory Conference (ICLTC'13)*, 2013.
- [13] W. Johnson, P. Rizzo, W. Bosma, S. Kole, M. Ghijsen, and H. van Welbergen. Generating socially appropriate tutorial dialog. In *ISCA Workshop on Affective Dialogue Systems*, pages 254–264. Berlin, Heidelberg, 2004.
- [14] M. Kallmann. *Object Interaction in Real-Time virtual Environments*. PhD Thesis, Swiss Federal Institute of Technology – EPFL, Lausanne, Switzerland, 2001.
- [15] A. Kokane, H. Singhal, S. Mukherjee, and G. Reddy. Effective e-learning using 3D virtual tutors and webRTC based multimedia chat. In *International Conference on Recent Trends in Information Technology (ICRITIT)*, pages 1–6, 2014.
- [16] R. Koper and R. van Es. Modelling units of learning from a pedagogical perspective. *Online Education Using Learning Objects*, 40:40–52, 2004.
- [17] S. Kopp, J. Allwood, K. Grammer, E. Ahlsen, and T. Stocksmeier. Modeling embodied feedback with virtual humans. In *Modeling communication with robots and virtual humans*, pages 18–37, 2008.
- [18] S. Kopp, B. Krenn, S. Marsella, A. N. Marshall, C. Pelachaud, H. Pirker, K. Thórisson, and H. Vilhjálmsón. Towards a common framework for multimodal generation: The behavior markup language. In *Proceedings of IVA*, volume 4133 of *LNCS*, pages 205–217. Springer, Berlin Heidelberg, 2006.
- [19] M. Limniou, D. Roberts, and N. Papadopoulos. Full immersive virtual environment CAVE[TM] in chemistry education. *Computers & Education*, 51(2):584–593, 2008.
- [20] L.-P. Morency, I. de Kook, and J. Gratch. Predicting listener backchannels: A probabilistic multimodal approach. In *8th International Conference IVA*, volume 5208 of *LNCS*, pages 176–190, 2008.
- [21] R. Niewiadomski, E. Bevacqua, M. Mancini, and C. Pelachaud. Greta: an interactive expressive eca system. In *8th International Conference AAMAS*, pages 1399–1400, 2009.
- [22] J. Rowe, S. McQuiggan, B. Mott, and J. Lester. Motivation in narrative-centered learning environments. *Proceedings of the workshop on narrative learning environments, AIED*, pages 40–49, 2007.
- [23] E. Wenger. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos, California, 1987.
- [24] V. Yngve. On getting a word in edgewise. *Papers from the Sixth Regional Meeting of the Chicago Linguistic Society*, pages 567–577, 1970.