# Traveller: An Interactive Cultural Training System controlled by User-Defined Body Gestures

Felix Kistler[1], Elisabeth André[1], Samuel Mascarenhas[2], André Silva[2], Ana Paiva[2], Nick Degens[3], Gert Jan Hofstede[3], Eva Krumhuber[4], Arvid Kappas[4], and Ruth Aylett[5]

[1] Human Centered Multimedia, Augsburg University,
Universitätsstr. 6a, 86159 Augsburg, Germany
[2] INESC-ID / Instituto Superior Técnico,
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal
[3] Wageningen University,
Hollandseweg 1, 6706KN Wageningen, Netherlands
[4] Jacobs University,
Campus Ring 1, 28759 Bremen, Germany
[5] Heriot-Watt University,
EH14 1AS, Edinburgh, United Kingdom

{kistler,andre}@informatik.uni-augsburg.de
{samuel.mascarenhas,andre.silva,ana.paiva}@gaips.inesc-id.pt
{nick.degens,gertjan.hofstede}@wur.nl
{e.krumhuber,a.kappas}@jacobs-university.de
{r.s.aylett}@hw.ac.uk

**Abstract.** In this paper, we describe a cultural training system based on an interactive storytelling approach and a culturally-adaptive agent architecture, for which a user-defined gesture set was created. 251 full body gestures by 22 users were analyzed to find intuitive gestures for the in-game actions in our system. After the analysis we integrated the gestures in our application using our framework for full body gesture recognition. We further integrated a second interaction type which applies a graphical interface controlled with freehand swiping gestures.

**Keywords:** User Defined Gestures, Kinect, Full Body Tracking, Depth Sensor, Interaction, Interactive Storytelling, Cultural Training

## 1 Introduction

Experience-based role play with virtual agents offers great promise for social training. In this paper, we present a system called Traveller (Train for Virtually Every Locality) that makes use of such an approach to educate young adults (18-25) in cultural sensitivity. The system implements a virtual storytelling environment in which the users learn by finding out how to appropriately interact with characters simulating different cultures as defined by Hofstede [5].

It is vital to the success of experience-based learning with virtual characters that the user is able to interact with them in a socially believable manner. As depth cameras have become broadly available with the Microsoft Kinect[1], we decided to make use of novel full body interaction techniques that allow trainees to practice culturally-varying non-verbal behaviors directly. However, the identification of intuitive gestures that are expressive enough to enable meaningful interaction with virtual characters is a challenge for the interaction designer. Given that gesture sets are usually chosen by the developers themselves, they do not necessarily have to be intuitive for the majority of users.

An approach that employs a user-defined gesture set has been presented by Wobbrock et al. for surface computing [14], and was adapted for other areas, such as public displays [9] or human robot interaction [11]. Its basic idea is to show specific effects within a system to users, who are then asked to perform gestures that should trigger these effects. The gesture performances are recorded and later analyzed to find gesture candidates that represent the choice of a majority of users. We adopted this approach for Traveller, in which full body gestures performed by the users trigger in-game actions that can vary in their type. For the present purpose we focus on two common action types: a) *navigation*, i.e. changing the position and perspective of the virtual camera and b) *dialogue* with virtual agents.

In the next section, we describe the scenario, including the story script and learning goals of our system, and the architecture used to model the agents' behavior. In Section 3, the development of the user-defined gestures set is depicted, which represents the main part of the interaction within our system. Afterwards, a secondary interaction type for cases in which the main interaction is hard to apply is presented, followed by a conclusion.

## 2 Scenario

### 2.1 Background Story and Learning Goals

In Traveller, users take the role of a character that has never been abroad. Throughout the game, they follow in the footsteps of their deceased grandfather, who used to travel the world. Their grandfather has left a letter, to be opened on the character's 18th birthday, in which he states that he has hidden a treasure long ago. To keep it safe, he has left pages of his travel journal in a few countries that he used to visit. To find these pages, the users have to travel to three different countries, and interact with the inhabitants of those countries in so-called critical incidents. The journal pages tell the users where to go to next, and also describe their grandfather's experiences as a beginning traveller.

The story starts at their grandmother's café, where the grandmother gives initial instructions. Afterwards, the users travel to the first country, in which they have to get directions from a group of strangers in a bar (first critical incident), have to find the responsible supervisor in a nearby museum in order to receive entry permission for a

[1] http://www.xbox.com/KINECT

park (second critical incident), and support or blame the supervisor when he knocks over a priceless artifact (third critical incident). In the second country, the users have to interact with an elderly man who wants to sit in the users' seat, interact with a train conductor who claims they have the wrong ticket, have a stranger join them for dinner and who might steal their wallet, and help out at a café to earn some money. In the third country, the users have to decide whether they care more about finding the treasure, or helping somebody in need, and interact with people at a party. At the end of the game, the users discover that the hidden treasure was actually in the experiences and adventures that they had during their travels.

The characters in each country have different rules for behavior and interpretation, depending on their synthetic culture. Therefore, if users do not select the appropriate actions, the virtual characters can get upset, and be unwilling to help them on their way. This can sometimes lead to obvious misunderstands, or even outright conflicts, but also to situations in which the users do not realize that they have offended the characters. The cultural configuration of the virtual characters is based on synthetic cultures [5]. These synthetic cultures reflect the extremes of Hofstede's dimensions of culture [4], which were empirically validated across several nations. At the moment, we only use three dimensions to determine the behavior of the characters: *power distance*, *individualism* vs. *collectivism*, and *masculinity* vs. *femininity*.

The aim of Traveller is comprised of an affective goal, which focuses on the users' emotions, and a cognitive goal, which focuses on the users' knowledge and understanding. The *affective goal* focuses on making the users aware that their rules for interpretation of appropriate behavior might be incorrect. For example, in certain cultures, women are not willing to talk to a stranger. As a stranger, it is easy to think they are just rude, but it might just be that it is inappropriate for women to talk to a stranger in public. Self-reflection and articulation of how the other person's behavior makes one feel then allows the users to recognize their emotions as they arise in reaction to such a novel situation. These emotional responses may not always be positive in the first place, but should be accepted and integrated without feelings of prejudice. The *cognitive goal* focuses on making the user understand general differences in cultures. As there are similar actions that occur within each country, the users are able to see the effect of similar actions in different countries. For example, a stranger would be treated differently in a collectivistic culture, than in an individualistic culture. By experiencing similarities and differences between cultures, the users see how various standpoints of cultural groups lead to respective behavior and assumptions. The attempt to see the same action from the point of view of people from another culture in turn prompts perspective taking and provides the foundation for empathic responses.

### 2.2 Cultural Agent Architecture

The reasoning and behavior of the characters in Traveller is driven by the Social Importance Dynamics (SID) model [10], which was integrated in the FAtiMA agent architecture [1]. The SID model is an adjustable model of cultural influences in social behavior that is based on Kemper's status-power theory [6]. The model augments the standard BDI agent framework [3] with a set of social dynamics that constitute human

socio-cultural behavior. More specifically, it models the human notion that others, from a relational perspective, are more or less important and that importance determines how much we are willing to act in their best interests. Conversely, it also determines how much we feel entitled to have others act in our favor. To give an example, if a close friend or a family member needs a place to stay overnight, we would gladly provide a room for him or her to sleep. However, the same is less likely to happen if the person is a stranger. Still, if the stranger simply asked for directions to a hotel, then most of us would comply.

Cultures greatly differ on how much social importance (SI) is attributed to others and how much it is conveyed by certain actions. For instance, in many Western cultures, cheek kissing is a very common greeting between acquaintances of the opposite sex. Oppositely, in China it is considered to be a very intimate act. The SID model enables the representation of such conventions. To illustrate how the model has been applied in Traveller, we consider the second critical incident of the first country that takes place in a museum, and in which the user is trying to find a supervisor of a wild park to ask for entry permission. If the user chooses to approach the supervisor directly, the supervisor's response will depend on his cultural configuration. One dimension of this configuration is *power distance*, which indicates how people treat others with higher status. If the supervisor's culture has a high *power distance*, he will not accept a direct request from the user with lower status. If the *power distance* score is low instead, the supervisor will directly accept the request.

Besides the SID model, the architecture also features the capacity to synthesize emotions in response to events that happen in the virtual environment, following the OCC appraisal theory [12]. This is essential for making the characters seem believable and for the user to establish an empathic relation with them. The architecture has also two different layers to control the behavior of the agents, a Reactive layer and a Deliberative Layer. The first one is responsible for generating quick reactions to events, such as a facial expression triggered by an emotion. The second one endows agents with goal-oriented behavior. These capabilities were already existent in FAtiMA and more details about them are described in [2].

The virtual environment for the scenario was implemented using the cross-platform game engine Unity3D[2]. The ION framework [13] is used to manage the communication between the agent architecture and the 3D world. In addition, characters speak by using the Microsoft Speech API and voices from CereProc[3].

## 3 Development of User-defined Gestures

For the development of our user-defined gesture set, we conducted a study that is described in the following section. In the subsequent section, we present the resulting gesture set and how we integrated it into our application.

---

[2] http://unity3d.com

[3] http://www.cereproc.com

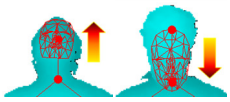
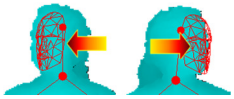
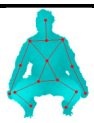
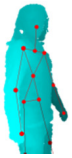
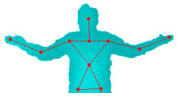
### 3.1 Gesture Study

We conducted our gesture study for the introduction scenario and the first two critical incidents within the first country. They included the following in-game actions to be triggered via body gestures: *yes, no, sit at bar and wait, approach group, ask for directions, leave the bar, ask about supervisor, ask guard to talk to supervisor, approach supervisor, ask permission.*

Whenever an interaction was requested by the system, we displayed all available actions as text boxes on the screen and the users' task was to invent and perform a gesture for each displayed option, one after the other. Those gesture performances were recorded on video and analyzed later to find gesture candidates for the investigated actions. The gesture candidates were chosen according to an agreement score based on how often users performed the same gesture for one action. In this way, we got one or two gesture candidates per action depending on the level of agreement between the participants. In the case of two gesture candidates we needed to decide which one we would use in the final gesture set. This was done in a way that the single gestures fitted to each other, and there was no problem of ambiguous gestures in parallel.

The final gesture set and its integration is described in the next chapter. Further details about the gesture study, the results of its analysis, and the implementation of the gesture recognition can be found in [7].

**Table 1.** Implemented gesture candidates and related actions (in brackets).

| | | | | |
|---|---|---|---|---|
|  head nod *(yes)* | |  head shake *(no)* | |  sit down *(sit at bar and wait)* |
|  step forward *(approach group)* |  turn away *(leave the bar)* |  point to front *(ask guard to talk to supervisor)* |  tip on shoulder *(ask permission)* |  arms out *(ask for directions, ask about supervisor)* |

### 3.2 Gesture Set and its Integration

For integrating the gestures in our application we use the Kinect for Windows SDK[4] together with our FUBI framework of which an earlier version has already been presented in [8]. FUBI achieves gesture recognition by using an XML-based definition language. To instruct users, we display symbols (single images or animations) that visualize how the gestures for these actions should be performed. The gesture candi-

[4] http://www.kinectforwindows.org

dates we extracted in our study and integrated in the first part of our scenario are depicted by their onscreen symbols in Table 1. Most of them are actually animations, but only one or two important key frames are displayed for reasons of clarity. As soon as a symbol is displayed on screen, the recognition framework automatically checks the corresponding recognizer for the closest user in the depth sensor's field of view. In case the recognition has been successful, it triggers an event related to the symbol in the same way as it is done for default interface buttons in Unity3D. Fig. 1 depicts a scene of the first critical incident displaying four symbols of the new gesture set.


**Fig. 1.** Gesture symbols in the first critical incident.

## 4 GUI Interaction

During the further development of Traveller's story, it became clear that our scenario would include interactions that sometimes have multiple conversational actions in parallel that would be hard to represent with unambiguous gestures. Therefore, we decided to add a second type of interaction to our application.

In case many conversational actions are necessary at a specific point in time, we group those actions into a dialogue menu as shown in Fig. 2. For entering the dialogue menu, users have to perform the "arms out" gesture as depicted in Table 1. When the dialogue menu is available, the symbol for the "arms out" gesture is shown in parallel to the other currently available actions, which are directly represented by gesture symbols. When the users perform the "arms out" gesture, the dialogue menu opens with the additional available conversational actions and one option to close the menu again. Within the menu, the options are arranged around a circle in the middle of the screen, with each of them occupying an equally sized sector around the middle circle. For selecting one of the options in the dialogue menu, users first have to stretch out

their hand to the front, wait until the menu gets activated, and then perform a swiping gesture in the direction of the option they would like to select. Activation of the menu is visualized by the middle circle changing its color from blue to yellow. In addition, the circle always contains textual instructions for what to do next. As soon as the start of a swiping gesture is recognized, the corresponding arrow gets a little stretched in its pointing direction, and also changes its color to yellow together with the background of the corresponding action text. As soon as the swipe is completed and thus a selection is performed successfully, a sound is played for additional feedback.

In this way, we keep the freedom to develop the story with as many and complex actions as we want, without worrying about how all of them could be represented by unambiguous gestures. However, the interaction modality remains the same for all in-game actions, and the two interaction types are similar enough to provide a fluent user experience.


**Fig. 2.** GUI Interaction menu

## 5 Conclusion

In this paper, we presented a novel approach to culture training that is based on role play with virtual characters. By engaging in interactions with characters that simulate different synthetic cultures, users may actively experience the challenges of cultural communication. Particular emphasis was given to the design of natural forms of gesture-based interaction to achieve the required social immersion which we consider as a decisive factor of success for social learning. First informal studies with the system at public events have shown that users are very engaged with the system, eager to explore the scenarios and enjoyed the gesture-based interaction. Future user

studies will investigate to what extent interaction with the system may foster cultural awareness.

**Acknowledgments.** This work was funded by the European Commission within FP7 under grant agreement eCute (FP7-ICT-257666).

## 6 References

1. J. Dias, S. Mascarenhas, and A. Paiva. Fatima modular: Towards an agent architecture with a generic appraisal framework. In *Proc. of the Int. Workshop on Standards for Emotion Modeling*, 2011.
2. J. Dias and A. Paiva. Feeling and reasoning: a computational model for emotional agents. In *Proc. EPIA 2005*, pages 127–140. Springer Berlin / Heidelberg, 2005.
3. M. P. Georgeff and F. F. Ingrand. Decision-making in an embedded reasoning system. In *Proc. IJCAI 1989*, pages 972–978, 1989.
4. G. Hofstede, G. J. Hofstede, and M. Minkov. *Cultures and organizations: Software of the mind: Intercultural cooperation and its importance for survival*. McGraw-Hill, New York, third edition, 2010.
5. G. J. Hofstede and P. Pedersen. Synthetic cultures: Intercultural learning through simulation games. *Simulation & Gaming*, 30(4):415–440, 1999.
6. T. Kemper. *Status, power and ritual interaction: a relational reading of Durkheim, Goffman, and Collins*. Ashgate Publishing Limited, England, 2011.
7. F. Kistler and E. André. User-defined body gestures for an interactive storytelling scenario. In *Proc. INTERACT 2013*, 2013.
8. F. Kistler, B. Endrass, I. Damian, C. Dang, and E. André. Natural interaction with culturally adaptive virtual characters. *Journal on Multimodal User Interfaces*, 6:39–47, 2012.
9. E. Kurdyukova, M. Redlin, and E. André. Studying user-defined iPad gestures for interaction in multi-display environment. In *Proc. IUI 2012*, pages 1–6, 2012.
10. S. Mascarenhas, R. Prada, A. Paiva, N. Degens, and G. J. Hofstede. Can i ask you a favour? - a relational model of socio-cultural behaviour. In *Proc. AAMAS 2013*. Springer Berlin / Heidelberg, 2013.
11. M. Obaid, M. Häring, F. Kistler, R. Bühling, and E. André. User-defined body gestures for navigational control of a humanoid robot. In *Social Robotics*, volume 7621 of *Lecture Notes in Computer Science*, pages 367–377. Springer Berlin / Heidelberg, 2012.
12. A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, UK, 1988.
13. M. Vala, G. Raimundo, P. Sequeira, P. Cuba, R. Prada, C. Martinho, and A. Paiva. ION framework – a simulation environment for worlds with virtual agents. In *Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 418–424. Springer Berlin / Heidelberg, 2009.
14. J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-defined gestures for surface computing. In *Proc. CHI 2009*, pages 1083–1092, New York, NY, USA, 2009. ACM.