# Autonomous Agents and Avatars in REVERIE's Virtual Environment

Fons Kuijk, CWI, Amsterdam, Netherlands
Konstantinos C. Apostolakis, Petros Daras, CERTH, Thessaloniki, Greece
Brian Ravenet, TPT, Paris, France
Haolin Wei, David S. Monaghan, DCU, Dublin, Ireland

## Abstract

In this paper, we describe the enactment of autonomous agents and avatars in the web-based social collaborative virtual environment of REVERIE that supports natural, human-like behavior, physical interaction and engagement. Represented by avatars, users feel immersed in this virtual world in which they can meet and share experiences as in real life. Like the avatars, autonomous agents that may act in this world are capable of demonstrating human-like non-verbal behavior and facilitate social interaction. We describe how reasoning components of the REVERIE system connect and cooperatively control autonomous agents and avatars representing a user.

**CR Categories:** I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence — Intelligent agents, I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism — Virtual reality;

**Keywords:** gaze, immersion, interaction, navigation, virtual environment

## 1 Introduction

Tele-immersion enables individuals that are geographically apart to interact naturally with each other in a shared 3D synthesized environment in which each of the participants can be represented by a virtual character, a so-called *avatar*. Not being restricted to the recordings of static cameras, tele-immersion systems can show participants what can be seen in the virtual environment from the dynamically changing point of view of their virtual representation. The work presented in this paper is part of the EC 7th Framework Programme project REVERIE [REVERIE 2011] that aims at developing such an immersive environment: an ambient, content-centric Internet-based environment where people can work, meet, participate in live events, socialize and share experiences as they do in real life, but without time, space and affordability limitations. In this environment we also may find real-time 3D video representations of users that we call *replicants* and virtual humanoids known as *agents*. Agents are autonomous; they do not represent a human user in the real world but act like one.


Web3D '15, June 18 - 21, 2015, HERAKLION, Greece

ACM 978-1-4503-3647-5/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2775292.2775305

In this paper we describe how modules of the REVERIE system running on web-connected client systems control autonomous agents, analyze captured data from the real human and model the interaction between virtual and real, including facial expressions, gaze and gestures as they are essential aspects of non-verbal communication. Related work is presented in Section 2. Section 3 is an overview of the modules that can be configured to run on individual user clients. Section 4 describes the behaviour of agents that can be obtained by the system and Section 5 describes how the system can support avatars. Conclusions are drawn in Section 6.

## 2 Related Work

The framework we developed for REVERIE is to some extent related to the framework of SEMAINE: a multimodal dialogue system that controls a virtual agent that interacts with humans and reacts to the user's non-verbal behavior [Schröder M, 2010]. That system is focused on facial expressions and gestures of the upper body. The interaction it supports is interaction between a human in the real world and a single virtual character, a Sensitive Artificial Listener, in a separated virtual world. In contrast to that, in the REVERIE system multiple participants and autonomous virtual humans share the same virtual space, which adds to the level of interaction and to the sensation of being immersed. The components of our framework that deal with behavior have to seek, identify, and process influencing factors in real-time to ultimately realize plausible behavior.

We can find related systems in which human users are represented in a virtual environment. In [Granieri et al., 1995] a system is presented to control behaviour of an interactive humanoid, and [Yoshida et al., 1995] presents a system that combines hand gestures and verbal descriptions for interacting with graphics objects. The VLNET (Virtual Life Network) system [Capin et al., 1995], provides a realistic representation through the use of motor functions, combined with interaction with the environment. Our REVERIE system differs from those systems in that its virtual environment can handle different representations of human users (semi autonomous avatars, puppeted avatars and replicants) and related to the requirements and resources available the level of control of avatars may vary: ranging from giving a rough impression on user behavior only, up to a level where it accurately mimics user behaviour (puppeting the avatar).

## 3 REVERIE's Configurable Clients

The REVERIE system is a distributed architecture of web-connected clients. Each of the clients can be configured to run the components needed for a particular application. In Figure 1we see

two such clients, one configured as a server just for reasoning and navigation and one configured as a user client that has a complete set of modules for user analysis. User clients in particular need a configuration of components customized for the application at hand. Communication between modules of the REVERIE system is done by means of both shared global parameters and publish/subscribe messages.

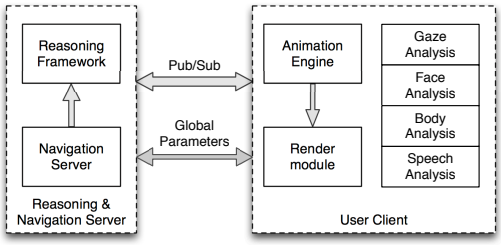


**Figure 1:** *Two web-connected REVERIE clients. On the left a client configured as a server for reasoning and navigation only, on the right a user client configured to analyze gaze, face, body and speech. Each user participating in the application has a client system appropriately configured for a specific application.*

A REVERIE constellation should have only one navigation module and one reasoning module. These components can run on a separate "server" client, but they may even run on one of the user clients as desired. Each client system should be configured to run an animation engine and a render component at least. Other modules are optional. In the next we will briefly describe the modules relevant for control of the virtual characters that can be configured to run on a client.

## 3.1 User Analysis Modules

Avatars represent users in the real world. For this there are modules that analyze user behaviour in the real world. There are modules to analyze *body gestures, facial expressions* and *speech* and modules to analyze *emotional and social* aspects.

### *3.1.1 Body gestures*

*Kinect Navigation*

A Kinect Navigation module provides the user with the ability to interact with REVERIE's Graphic User Interface (GUI) by interpreting different user gestures. In this way the user has the option to interact with the system without need for keyboard and mouse interfaces, intended to provide a natural and immersive experience. The workflow is shown in Figure 2. Once a user walks into the scene, the OpenNI skeleton tracking system automatically "picks up" (i.e. tracks) the user within 1 to 2 seconds. After that, the user can control settings and navigate through the world using a set of gestures. As the OpenNI component can track two skeletons at the same time, the first user that walks into the scene will have control of this module.

*Kinect Body Capturing*

A user can participate in the virtual environment as a body puppeted avatar. For this, the user's system should be configured to activate the Kinect Body Capturing component of a Puppeting module (see Section 5.1). This module is responsible for creating an ALM channel and publishing data to it. Each skeleton capturing frame is translated to a list of 17 4x4 rotation matrices corresponding to tracked rotations of the 17 OpenNI joints, and a similar list of 3D tracked positions for each joint in the hierarchy. The module proceeds to create a data buffer out of this information, which is published on the capturer's ALM channel.

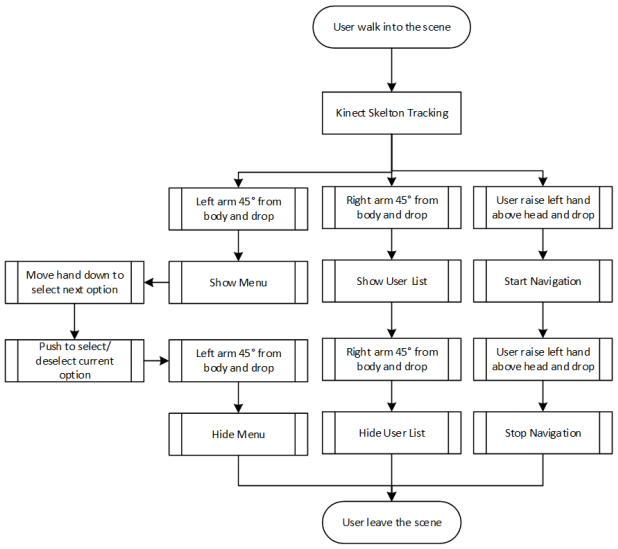


**Figure 2:** *Workflow of the Kinect Navigation module.*

### *3.1.2 Facial expressions*

Similar to the Kinect Body Puppeting Module, a Webcam Face Puppeting module allows users to control the movements of their avatar's face through a direct mapping of the character's face mesh geometry to a number of tracked feature points on the user's face (see Section 5.2). The latter are obtained through the deployment of webcam-based feature point tracking, which capitalizes on Active Shape Models (ASMs) [Cootes, 1995]. Correspondences of the 2D feature points in the image plane and the avatar's face mesh 3D vertices are drawn in order for the character rendering buffers to be updated. The resulting rendered frame depicts the character's face "mimicking" the user's expression (see Figure 12).

### *3.1.3 Speech analysis*

The role of a module for keyword recognition is to provide the virtual agents with a means to understand the speech of users. For our applications we just needed this module to be capable of recognizing different ways of saying "yes" or "no" and numbers. However its dictionary can be programmed to have more options. The module is based on the CMU Sphinx library [Lamere, P et al. 2003]. CMU Sphinx provides a powerful open solution for speech recognition and the library does not need a live Internet connection. This component is used in the different scenarios of REVERIE to trigger the virtual agents' reactions.

### *3.1.4 Emotional and social analysis*

The REVERIE Human Affect Analysis Module consists of three components: *User Affect Recognition*, *Head Nod and Shake Detection* and *Gaze Direction & User Engagement* as shown in Figure 3.

*User Affect Recognition*

The User Affect Recognition module gives a continuous predication on the arousal and valence level. Accurate face landmark detection is an important step of user affect recognition.

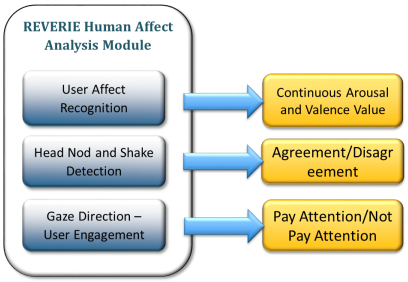


**Figure 3:** *Overview of REVERIE Human Affect Analysis Module*

We use the work proposed by [Xiong et al. 2013] to track 46 landmarks of the face (Figure 4). The normalized coordinates of each landmark are used as the input features. The Support Vector Regression (SVR) technique is used to learn the relationship between the input features and arousal and valence levels. To train the model we used the training and developing dataset from the Audio/Visual Emotion Challenge (AVEC) 2012 [Schuller, Björn, et al 2012].

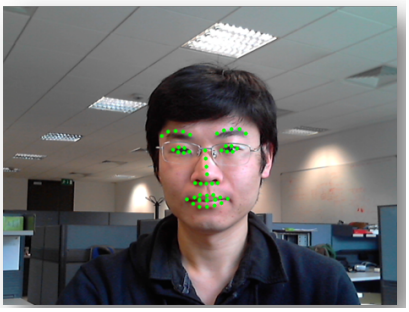

**Figure 4:** *Facial landmark detection.*

*Head Nod and Shake Detection*
The Head Nod and Shake detection module identifies head gestures as an indication of agreement or disagreement. Head nod and shake is detected by counting head movement direction between two adjadcent frames in a certain period of time.

*Gaze Direction*
The Gaze Direction module can observe where the user is looking at and use that as an indication of level of engagement. Engagement is determined by assessing whether or not the user is looking at the screen, based on head orientation. The method uses a threshold measure of the head pose to determine attention. When the user's head pose varies within the range, the user is considered to be engaged. This approach is less accurate compared to a more precise gaze tracking approach, however, it is very robust to head pose variation and does not require calibration.

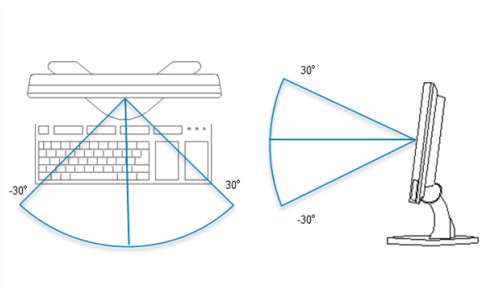


**Figure 5:** *Head pose thresholds*

## 3.2 Reasoning Module

Virtual characters that act in REVERIE's virtual environment can be autonomous agents, avatars representing a user and so-called replicants (a real-time 3D video representation of a user). Agents are completely autonomous, fully controlled by the *Reasoning Framework*. For the non-puppeted avatars, the Reasoning Framework controls gaze, gestures and pose based on user interaction.

Agents can be programmed on what to do by means of a use case specific script. A script can describe how the agent should behave, what to say when and where, what to ask whom, where to go to, etc. A script has options for flow control, so it can specify a non-linear story. User input in particular may influence execution of script instructions. An agent can have a dialog with one user or with a group of users. For simple dialogues (e.g., agree or disagree) both spoken input and user head movements (nod and shake) are accepted. The Speech Recognition Module supplies information on spoken input; the User Affect Analysis Module supplies head nod and shake input. For more complex dialogues spoken input may be a requirement. When the agent addresses one user, it will accept the first reaction that is recognized as being a valid answer. When the agent addresses a group – for instance when a tour guide asks a group of students if they appreciated the tour – the agent will wait a moment to allow all group members to respond and only then interprets the valid responses given to deduce, based on a simple majority rule, what he assumes to be the answer of the group as a whole.

Part of a script can be additional information on the environment, for instance on points of interest such as specific locations and position of seats. This type of metadata on the environment is used for navigation, gaze and pointing.

A script may contain scenarios that describe context-specific behaviour of the agent. As example, scenarios may describe how to react on users not being engaged, or how to react towards the winner of a game. Scenarios may come in different versions to avoid repetitive behaviour.

While it is important to enhance the animations of the agents in order to improve their realism, it is also important to enhance their believability by adapting their behaviours appropriately to the context. If several virtual characters behave identically while interacting with each other, it does not look realistic. People can exhibit different attitudes and this impacts their nonverbal behaviour. When communicating, a sentence can be expressed in a very different manner depending on the attitude to be conveyed. A computational model of nonverbal behaviour for the virtual agents of the REVERIE system has been proposed that allow an agent to convey a socio-emotional attitude [Ravenet et al. 2013]. By changing the attitude an agent expresses, the system outputs the animation parameters needed to convey that attitude. We can then change the generated nonverbal behaviour accordingly, generating different behaviour styles. Creating these various behaviour profiles, we create richer worlds with virtual characters able to behave in their own style.

The model we developed produces animation parameters, précising if gestures are activated, describing the gesture shapes, the head orientation and facial expression the agent should have and if the agent should avoid mutual gazes. The model has been extended to group situations in [Ravenet et al. 2014]. This model is used in the REVERIE system to generate profiles of different agents: a dominant one and a friendly one (see Figure 6).

Animation parameters are retrieved from the outputs of the model and animation for the agents are designed following these recommendations.

**Figure 6:** *Example of two different attitudes. Left: Dominant. Right: friendly*

## 3.3 Navigation Module

All characters need support for moving around in the environment, not only to avoid collisions with static objects such as walls and furniture; they also have to avoid colliding with each other. This is the major responsibility of the *Navigation Module*.

The Navigation Module operates on basis of the Explicit Corridor Map (ECM) method [Geraerts, R., 2010]. An indicative route is created that determines the global route for the character from the current position towards a new destination. A corridor around this route is used to handle a broad range of other path planning issues, such as avoiding characters and computing smooth paths. When the character is moving along the indicative route the dynamic collision detection is active. This collision detection component may decide on leaving the indicative route to avoid possible collisions with other characters or moving obstacles crossing the indicative route. If this should occur, the character may reduce speed and follows an alternative path around the obstacle, but still within the limits of the corridor that is determined by the static objects that make up the scene.

The Navigation Module continuously calculates and updates the current position and orientation of agents and avatars; at appropriate times it also triggers the Animation Module to initiate animations (like "walk", "rotate", "stand") and changes of pose (as "sitting"). It does so in such a way that the animation corresponds to the phase of the navigation action being performed.

**Figure 7:** *The Navigation Module knows about seat positions and can make a character sit down and stand up.*

Scene data can include data on seat positions. From navigation point of view, the location of a seat itself is inherently an inaccessible point that cannot be part of an indicative route. Therefore, the system does accept locations on or close to a seat location and interprets that as a signal: "I want to get seated". For calculating the indicative route, the inaccessible seat location is replaced with a seat access point, an accessible point near the seat. The navigation will bring the avatar up to this access point, turns the avatar in the seat direction, ready to be seated (shown in Figure 7) and generates a procedure to make the Animation System change the pose of the character from standing to sitting. When navigating away from this seated position, a stand-up procedure is started and the access point is taken as start of the indicative route.

### 3.3.1 Navigation of Agents

Agents are fully software controlled and, based on their script or reacting on an event, are instructed to go from one destination to another. For these agents the navigation system has one level of support only: fully autonomous path planning and collision avoidance.

### 3.3.2 Navigation of Avatars

For avatars, being the alter ego of a user, path planning and obstacle avoidance primarily serves to *assist* in navigation, giving users the opportunity to have a say in final positioning and orientation of their avatar. For this the navigation system supports users at different levels (see Section 5.6) and has the following options:

- *Keyboard navigation*. The system allows the user to navigate his avatar using the keyboard, taking small steps at a time.
- *Mouse navigation*. By pressing the right mouse button, the user can change speed and direction of the avatar when moving the mouse.
- *Kinect navigation*. A Kinect based gesture recognition system allows the user to navigate through the world using a set of gestures.
- *Map navigation*. The user can enter a new destination by clicking on a desired location in a top view image of the environment.
- *Follow-Me*. Participants can be part of a group; navigation of the system-controlled autonomous agent can initiate a Follow-Me mode that affects navigation of the whole group.

## 4 Result: Autonomous Agent Behaviour

In the above we have described the modules that play a role in the control of autonomous agents. In this section we will depict the result of this combined effort.

### 4.1 Reaction on (lack of) user engagement

The Reasoning Framework is signaled on the state of user engagement. This information is interpreted and may stimulate an agent reaction. If a user seems not engaged for a while, the agent may pick one of the scenario options available to call for attention. One scenario for instance can make the agent turn its gaze towards the inattentive user, wave with its right arm and call for

attention. The agent does so without moving from its place. A more intimidating reaction will make the agent walk to right before the inattentive user, look him in the eye, call for attention and return to its original position before continuing the show.

Not only the how, also when the agent will react is somewhat arbitrary. The agent will not call for attention immediately, and also, agents "learn" on users behaviour: by administering lack of engagement users that are frequently not engaged will in time be ignored.

### 4.2 Gaze

An agent has knowledge on the current position of all avatars in the environment. When the agent addresses a group it will be looking at individual members of that group alternatively, sometimes shortly sometimes a bit longer and skipping those that may be behind its back. Gaze is updated continuously, so the agent keeps having eye contact even if the agent and avatars are moving (see Figure 8).

**Figure 8:** *The agent's gaze towards the (moving) avatar that is being addressed.*

### 4.3 Playing Simon Says

The Reasoning Framework is familiar with the rules of a game known as "Simon Says". Based on this it can control an agent that leads that game autonomously. The agent optionally starts with an explanation of the game, and asks if the users are ready to play the game. When the users say so, the "Simon" agent will announce the start and will play the game. The agent selects one of the set of gestures in his repertoire, tells the players which one to perform either with or without the words "Simon says…", performs that gesture himself and next takes some time to analyze what the players are doing. The Reasoning Framework gets information from the User Activity Module based on which the agent draws conclusions (that is, decides on points the individual players score). One round of the game is doing this for several times. The Simon agent knows which of the players won a round of the game and will congratulate this player by addressing that player individually. Next Simon will ask if the players want to continue for a next round or stop the game.

The Reasoning Framework can handle a number of agents simultaneously. Simon can have an agent configured to be his assistant. We made this assistant agent join Simon in performing the gestures and at the end of a round the assistant agent will address the winner of the game.

## 5 Result: Avatar Interaction

Not only agents, also avatars are controlled by the system. In this section we describe how the user can control avatars directly (puppeting) but also how the modules that play a role in the control of autonomous agents can control them if puppeting is not available or not desirable.

### 5.1 Avatar Kinect Body Puppeting

*Puppeting* or *avateering* a virtual 3D avatar refers to the process of mapping a user's natural motoring activity and live performance to a virtual human's deforming control elements in order to faithfully reproduce the activity during rendering cycles. This process requires that the avatar mesh is parented to an articulated structure of control elements called *bones*, which are reminiscent of a human's skeletal structure. These *bones* can be viewed as oriented 3D line segments that connect transformable *joints* (such as a knee or shoulder). These joints usually offer a three-to-six Degrees-Of-Freedom deformation control of the avatar's mesh geometry with respect to translation and rotation transformations. In REVERIE, user avatars are rigged (i.e. parented) using a highly detailed hierarchy of joints. The original Kinect sensor on the other hand supports user skeleton tracking algorithms for up to 17 joints, defined in the OpenNI library. The REVERIE avatars created using external tools [Apostolakis, et. al, 2013b] provide a high-level abstraction of the joint hierarchy, resulting in mesh assets that are completely deformable via transformations applied directly on the Kinect-based 17 mesh joint counterparts.

The Avatar Kinect Body Puppeting module of the REVERIE system prototype is an end-to-end capturing and rendering module that generates animation based on skeleton data received over the network from a Kinect capturing station. The module is deployed using an integrated ALM server-client framework to transmit skeleton data over the network. The complete Kinect Body Puppeting module consists of a single capturer component deployed per user client (see Section 3.1.1), and multiple receiver components, which correspond to the total number of capturers identified within the network. Each capturer subscribes to its channel over the network and transmits Kinect skeleton data per frame, which is received by all receivers subscribing to the capturer's channel. The REVERIE system is responsible for distributing the correct parameters for each user to create the required number of receivers on the rendering side component of the module for rendering the amount of remote Kinect body-puppeted avatar participants. The architecture of the module is shown in Figure 9.

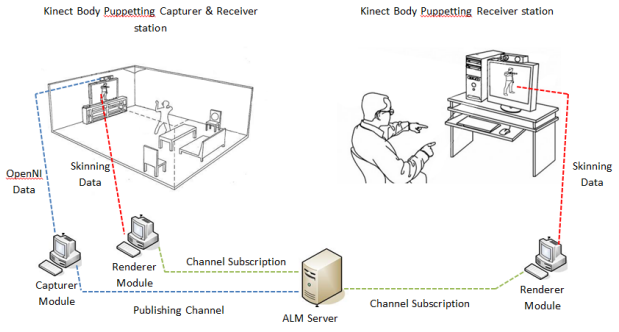


**Figure 9:** *Kinect Body-Puppeting module architecture*

The Rendering component of the Puppeting module serves a three-fold purpose:

a) it generates a number of ALM channel subscribers which receive and reconstruct the initial list of joint matrices and positions published by the corresponding capturers;

b) it loads the appropriate avatar assets from the system Cache; and

c) it handles the real-time rendering and animation of all body-puppeted avatars in the system.

The actual process of the puppeting algorithm from Kinect data and implementation thereof in the GPU via skinning shaders is similar to the scheme described in [Apostolakis, et. al, 2013a].

The Kinect Body Puppeting module makes extensive use of the Kinect sensor. Modules are able to share the Kinect's resources using a singular module for making calls to the OpenNI third-party libraries, called the *Shared Skeleton* module. The result of this coordinated effort can be seen in Figure 10.

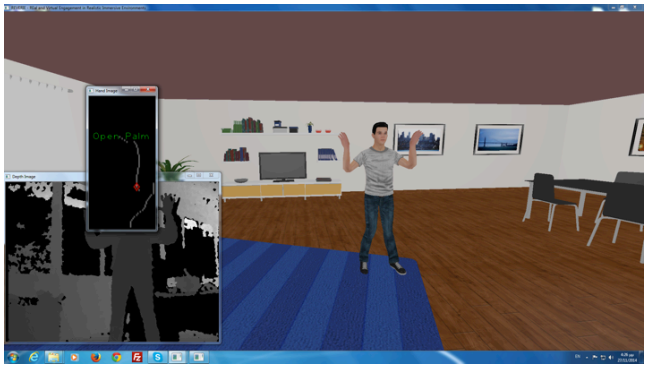


**Figure 10:** *Kinect Body-Puppeted avatar shown in the 3D Hangout environment using the Kinect gesture-driven navigation module via Kinect skeleton resource sharing offered by the Shared Skeleton common module.*

## 5.2 Avatar Webcam Face Puppeting Module

As is the case with the Kinect-based puppeting module, the core module for Webcam-based Face Puppeting of virtual humans is broken down into two components; one intended to perform the user face detection, tracking and feature point extraction from a single, front-facing web camera connected to the system (see Section 3.1.2), while deformation and rendering of the character's face mesh geometry happens in a separate component.

In order for the aforementioned components to communicate, a publish/subscribe ALM network infrastructure is deployed. In a similar manner to how the Kinect Body Puppeting Module utilizes the OpenNI library to access the Kinect skeleton capturing resources, the Webcam Puppeting capturer invokes a child process to open the user's web camera and gain access to the ASM tracking data by "listening" at the standard output channel. For each camera frame being tracked by the child process, the capturer generates a data buffer containing the image-plane coordinates of the 2D facial feature points and proceeds to publish the generated buffer to its designated channel.

In accordance, a remote Webcam Puppeting receiver subscribing to the capturer's channel is able to retrieve the original data published, and reconstructs a 3D representation of the ASM feature points by mapping correspondences of the image plane coordinates to the 3D vertices of the character's face mesh. The mesh renderable vertex data buffer data is updated with the reconstructed vertex information, and is then passed on to the rendering pipeline.

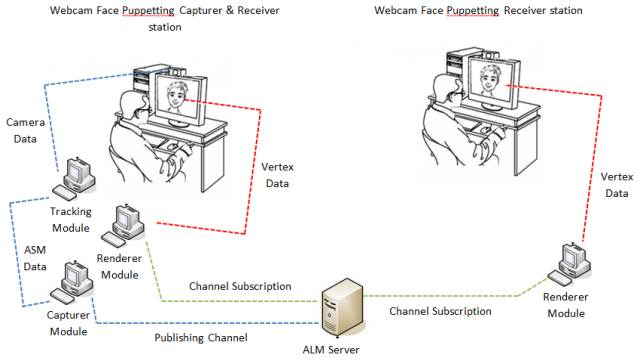


**Figure 11:** *Webcam Face-Puppeting module architecture.*

A diagram showcasing the module components is shown in Figure 11. For the actual avatar to be rendered on the receiver side, both the rendering component of the module as well as a Character Animation module have to be initialized together and load up the correct character assets. Through the export feature of the *Reverie Avatar Authoring Tool*, character assets are exported with a single mesh file representing the character's deformable face and rest of the body. Both assets are required to be loaded by each respective module in order for a complete character to be properly displayed in the system. Figure 12 depicts this complete character rendered within the REVERIE prototype system.

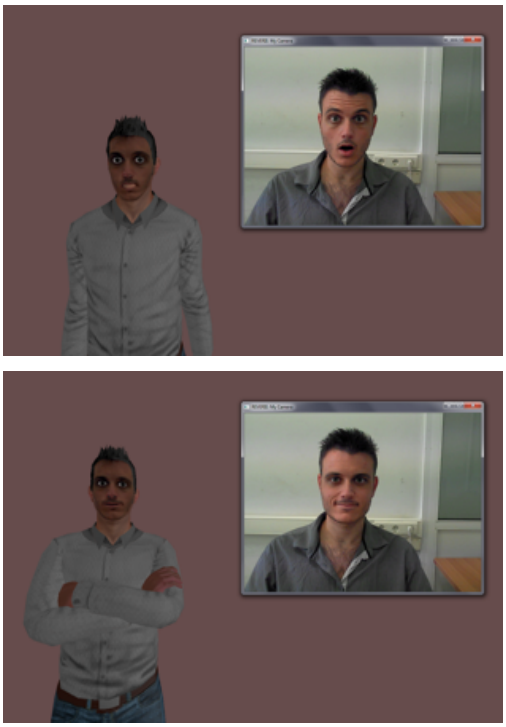

**Figure 12:** *Webcam Face-Puppeted avatar shown rendered within the REVERIE system. The user's current webcam retrieved frame is shown on the upper right of each image to showcase the face mesh deformation effect according to the currently tracked user expression.*

## 5.3 Engagement

The User Affect Analysis Module analyzes user engagement based on user attention. If the user is paying attention or not is detected by means of eye tracking and head orientation. By analyzing these user characteristics the module determines a level of engagement. The Reasoning Framework is informed on user engagement. This information is used to control the avatar gaze (users that are not engaged seem to look over their shoulder as shown in Figure 13). In this way participants are aware on the engagement of the other participants.

**Figure 13:** *The avatar's gaze reflects the user's engagement, in the left image the user is engaged, in the right image the user is not engaged.*

## 5.4 Camera Control

A user can control the orientation of the virtual camera that determines the view on the virtual environment. In first person and third person view the default orientation of the camera is aligned with the forward direction of the avatar. By using the mouse (left button) the user can change this camera orientation and look around in the environment from the position of the avatar. This looking around is limited roughly to angles that a human being in the real world can obtain by moving his head and eyes. When the user is manipulating the virtual camera, the avatar gaze follows this "camera gaze" (see Figure 14), so a participant can be aware of what other participants look at.

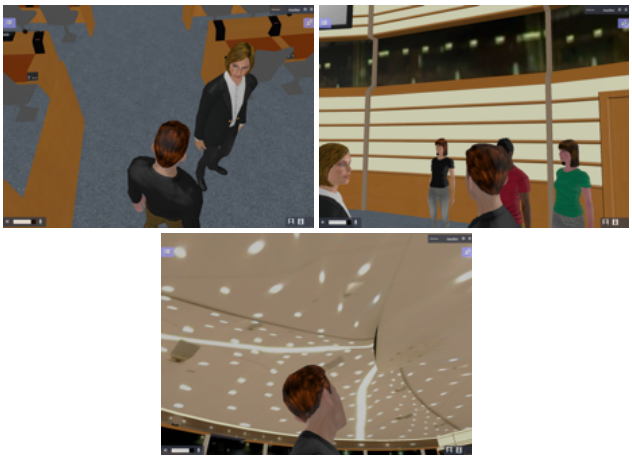

**Figure 14:** *The avatar's gaze follows the users camera setting.*

## 5.5 Request to Speak

In one of the scenarios we implemented, students may ask for permission to speak. They can do so by pressing a button. The Reasoning Framework makes this request visible: the student's avatar will raise its arm (see Figure 15). This request-to-speak state is limited in time. After an appropriate moment the arm will be lowered again and the student will have to reconfirm his request.


**Figure 15:** *A request to speak result in the user's avatar raising a hand.*

## 5.6 Navigation

Whereas agents navigate as their actions require, avatars are user controlled. Based on the configuration of their system a user may have the following options for manual navigation.

### *5.6.1 Keyboard Navigation*

Keyboard navigation is a step-wise form of user-controlled navigation that does not initiate calculation of an indicative route. The system checks if the small step to take in any of four directions (forward, backward, left and right) will not cause a collision; first with static obstacles, and next with dynamic objects. If this is not the case, the character is moved towards that new position in a smooth way, taking a specified number of frame-times. Keyboard navigation also allows users to rotate their avatar on the spot.

### *5.6.2 Mouse Navigation*

Mouse navigation is a continuous form of user-controlled navigation that does not initiate calculation of an indicative route. The system continuously checks if the moving avatar does not cause a collision and will stop the motion immediately if this would be the case. As mouse displacement has two dimensions, it allows change of both speed (mouse forward and backward) and direction (mouse left and right), which makes it possible to move along curved trajectories. It is the only form of navigation with which the user can make the avatar move backwards (in which case it triggers the "walk-backwards" animation).

### *5.6.3 Kinect Navigation*

If a system is properly equipped, a user can make use of the Kinect navigation feature (Section 3.1.1). Like mouse navigation it is a continuous form of navigation, and likewise, it does not

initiate calculation of an indicative route. Also here the system continuously checks if the moving avatar does not cause a collision and it stops the motion if this would be the case. Kinect navigation is added to the system to allow a user that does not have the ability to reach for a keyboard to move his avatar forward and change direction by means of gestures. A combination of these two makes the avatar move along curved trajectories.

### 5.6.4 Map navigation

Map navigation is a semi autonomous system-controlled form of navigation initiated when the user specifies a new destination by clicking on an image that shows the environment from above (see Figure 16). This modus operandi is helpful in particular for navigations full of twists and turns (long distance and/or complex situation). It is based on the full ECM method starting with calculation of the indicative route as described in the above. The system looks for the nearest accessible position if the user happens to click on an inaccessible point.

**Figure 16:** *Top view image for user initiated navigation*.

The system interprets a mouse-pressed location as destination, and the mouse-released location as an indicator of direction. In this way one click-drag-release input specifies both location and direction. When a direction is specified in this way, the navigation will not only bring the avatar to the requested position, but will make it turn into the specified direction after arrival.

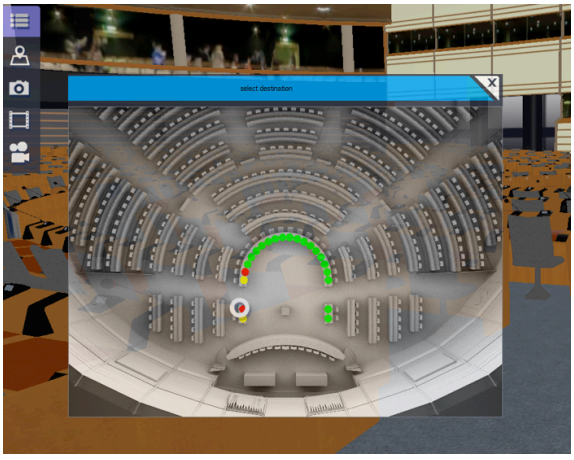


**Figure 17:** *The user interface shows seat options: free (green), occupied (red) and assigned (yellow).*

If scene data includes specifications of seats, these seat options are made visible in the map (see Figure 17). A green dot indicates a free seat, a yellow dot indicates the seat has been assigned (to a character still on its way) and a red dot indicates the seat is occupied. Clicking on a free seat indicator will make your avatar move to that seat and make it sit.

### 5.6.5 Follow Me

In Follow-Me mode, destinations of individuals of a group are assigned by the system based on the destination and final orientation of the agent (e.g., a tour guide) the group is following. Automatically assigned destinations are positions in front of the agent with a distribution density decreasing as distance increases. Examples of this automated mise-en-place are shown in Figure 18.

**Figure 18:** *Navigation of a group: the system automatically assigns positions in front of the agent based on user engagement and available space*.

As noted before, the REVERIE system has components that infer engagement of the user. The automatic assignment of positions is prioritized based on the level of engagement. As a result more engaged participants obtain positions more near to the agent than the less engaged participants. This level of engagement also influences the walking speed assigned to individual group members when the group as a whole is on the move.

In Follow-Me mode users can still manipulate their avatar, either by mouse navigation or by map navigation. In this way the user can try to obtain a better view, or to get near to a friend. In a future version we may support social aware assignment so placement near a friend gets priority.

## 6 Conclusions

The different components presented in this paper allow for the realization of human-like interactions within a web-connected virtual environment. The different scenarios of the REVERIE system include different specific recognition mechanisms of the user's activity. Thanks to non-intrusive real-time recognition processes, the speech, the gestures, the emotions and the facial expressions of the users are recognized by the system in a dynamic and natural fashion. This way, real-time puppeting of the avatars is made possible along with mouse-keyboard-triggered actions like navigation or requesting to speak. Depending on the activity of the users, the avatars produce rich human-like behaviors that are visually understood by the other users and the virtual agent. The virtual agent itself is capable of reacting autonomously to the users' activity in various ways, detecting when a user is losing engagement or participating in the interactions and showing appropriate behaviors (gestures, gaze, speeches) in response. The components can be deployed on user computers, as they do not require a heavy setup (Kinect, webcam and microphone) and therefore are well suited for a web-based communication system.

## Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. ICT-2011-7-287723 (REVERIE project).

The navigation system is build on top of the ECM library made available by the University of Utrecht, Department of Information and Computer Sciences.

## References

APOSTOLAKIS, KONSTANTINOS C., ET AL. 2013a "Blending real with virtual in 3DLife". *Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2013 14th International Workshop on. IEEE.

APOSTOLAKIS, K. C., AND P. DARAS. 2013b "RAAT-The reverie avatar authoring tool". *Digital Signal Processing (DSP)*, 2013 18th International Conference on. IEEE.

CAPIN, T K., PANDZIC, I, THALMANN, N., THALMANN, D., 1997, "Virtual Human Representation and Communication in the VLNET Networked Virtual Environments", *IEEE Computer Graphics and Applications*, Vol.17, No2, 1997, pp.42-53.

COOTES, TIMOTHY F., ET AL. 1995, "Active shape models-their training and application." *Computer vision and image understanding* 61.1 (1995): 38-59.

GERAERTS, R., 2010, Planning Short Paths with Clearance using Explicit Corridors. In *IEEE International Conference on Robotics and Automation*, pp. 1997-2004

GRANIERI J.P., BECKET W., REICH B.D., CRABTREE J., BADLER N.I. 1995. Behavioral Control for Real-Time Simulated Human Agents. *Proceedings of ACM Symposium on Interactive 3D Graphics*, Monterey, California.

LAMERE, P., KWOK, P., GOUVEA, E., RAJ, B., SINGH, R., WALKER, W., ... & WOLF, P. (2003, April). "The CMU SPHINX-4 speech recognition system". In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing* (ICASSP 2003), Hong Kong (Vol. 1, pp. 2-5).

PANDZIC, I., MAGNENAT THALMANN, N., CAPIN T., THALMANN D., "Virtual Life Network: A Body-Centered Networked Virtual Environment", Presence, MIT, Vol. 6, No 6, 1997, pp. 676-686.

RAVENET, B., OCHS, M., & PELACHAUD, C., 2013, January. From a user-created corpus of virtual agent's non-verbal behavior to a computational model of interpersonal attitudes. In *Intelligent Virtual Agents* (pp. 263-274). Springer Berlin Heidelberg.

RAVENET, B., CAFARO, A., OCHS, M., & PELACHAUD, C. 2014, January, "Interpersonal Attitude of a Speaking Agent in Simulated Group Conversations". In *Intelligent Virtual Agents* (pp. 345-349). Springer International Publishing.

REVERIE 2011, "REal and Virtual Engagement in Realistic Immersive Environments", http://www.reveriefp7.eu

SCHRÖDER M, 2010, The SEMAINE API: Towards a Standards-Based Framework for Building Emotion-Oriented Systems. *Advances in Human-Computer Interaction*.

SCHULLER, BJÖRN, ET AL. 2012. "Avec 2012: the continuous audio/visual emotion challenge." *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM

XIONG, XUEHAN, AND FERNANDO DE LA TORRE, 2013. "Supervised descent method and its applications to face alignment." *Computer Vision and Pattern Recognition (CVPR)*, IEEE Conference on. IEEE, 2013

YOSHIDA M., TIJERINO Y., ABE S., KISHINO F., 1995. "A Virtual Space Teleconferencing System that Supports Intuitive Interaction for Creative and Cooperative Work". *Proceedings of ACM Symposium on Interactive 3D Graphics*. Monterey, California.