Script-based Multimodal Output Generation for a Conversational Character

Andrea Corradini, Manish Mehta, and Niels Ole Bernsen

University of Southern Denmark, Natural Interactive Systems Laboratory (NISLab)
Campusvej 55, 5230 Odense M, Denmark
{andrea,manish,nob}@nis.sdu.dk

Abstract

This paper reports on the generation of coherent multimodal output of an embodied conversational character in a 3D game setting. In our implementation, we employ a hierarchical two-tiered approach that supports the designer in creating agent behaviors through scripts. Lower-level scripts define sets of both elementary animations and custom animations, along with their temporal duration. At a higher level, sequences of elementary or custom animations are used to define more complex behaviors of variable time duration. At each hierarchical level, additional scripts of syntactic rules take care of specifying smooth transitions from one animation and to the next one. The use of placeholders within scripts allows for parts of the non-verbal output to be selected at run-time thus increasing variety in behavior display. The English-like language chosen for the scripts makes the system very versatile and easy to use. The generation system works in real-time and any user can easily add behaviors at will.

1 INTRODUCTION

1.1 GAMES AND EDUTAINMENT

The games software industry is expanding rapidly. A recent poll conducted for the Entertainment Software Association shows that computer and video game software sales are expected to show strong growth over the next few years as consumers plan to keep spending on them. When inquired about the reasons, gamers say they play games because these are fun (87%), challenging (72%), are an interactive social experience (42%) and provide a lot of entertainment value for the money (36%) [7].

By deploying the increasingly powerful processing power and memory currently available in PCs and consoles, and by adopting more advanced AI strategies, games [5] are becoming more and more complex. Their scenarios are usually characterized by high-quality graphics and sound to both offer involving interactivity and present users with engaging challenges. Games require the user to apply different skills ranging from simple eyes/hand coordination to resource manipulation, strategic thinking and planning. Besides providing pleasure and challenge, play may also motivate the user thus increasing her involvement and potentially helping her learn[1] [31]. In fact, not only philosophers and social scientists [14, 22] relate play to sociological concerns and biological functions that have to do specifically with learning. Recently, also governmental agencies [1], responsible for investigating the role of technology in schooling and training, have recognized that games may contribute towards achieving educational objectives by establishing projects to look at aspects of games that might be of value to education.

1.2 INTERACTIVITY AND MULTIMODALITY IN COMPUTER GAMES

Especially in the game industry, there have been attempts to provide the user with different modalities and an enhanced user engagement in order to elicit a rich interaction experience. Back in 1983, Sierra On-Line (now Sierra Entertainment [9]) was asked by IBM™ to come up with a game that would take advantage of IBM™ PC's 16-color palette, three-channel sound, and 128K of memory. Within short time, a small team of programmers and artists designed the game *King's Quest I: Quest for the Crown* in which the player takes on the persona of Sir Graham, on a quest to recover three lost treasures. With its release, this game probably became the first three-dimensional, animated, interactive cartoon. The player could control the main character's movement and actions using the keyboard's arrow keys and by typewriting simple sentence commands, respectively.

[1] Other play activities and, with regard to media, TV shows such as Sesame Street support the belief that humans, notably children, are captivated by several forms of learning by playing

Black and White [2] is a strategy game developed by Lionhead Studios Ltd., which involves drawing symbols on the screen to cast miracles such as a rain spell to increase food production. The shape of the symbols ranges from arrows and swirls to letters and numbers. As the user advances in the game, the symbols she has to draw become more complex, making it harder to cast them quickly during gameplay.
Age of Empire [4] is a computer game in which hand-drawn and 3D rendered graphics aim to create an involving experience for the user. Employing multimedia output, the goal is to bring her gaming experience into focus. With Microsoft Speech API (SAPI) [3] speech commands can be used as additional input modality for the game. Similarly, *Life Line* [8] is an action-adventure game for Playstation 2, in which players can verbally guide the main character through the game. Both games feature a voice-activated command set rather than conversational dialogue therefore increasing the user's personal immersive experience. Yet, the effectiveness of speech is arguable because it only replaces the menu selection task provided by the game graphical interface with a large set of utterances defined beforehand. Instead of facilitating the interaction, this may potentially cause an increase in the user cognitive load because she has to memorize sequences of arbitrary commands. Furthermore, state of the art voice recognition technology is not as advanced as it needs to be to effectively support the entire gameplay experience.
Only few game systems have been put forward by academic and research organizations. *MIND-WARPING* [33] is a multi-player game for wearable computing and augmented reality environments. In the game, two players compete against one another. One player acts as martial art fighter, the other one as a wicked magician. This latter has an overhead view of the playfield to direct demos against the fighter. The fighter has to perform magical gestures and shouts to keep off the demons. She interacts with the game by voice commands and gestures captured via computer vision techniques in a first-person augmented reality.
Pirates! [12] is a collaborative multi-player computer game that attempts to maintain some of the social aspects of traditional game play by transferring some computational parts of the game into the real physical world. The mapping of physical world locations onto the game world dictates game events. Wireless hand-held devices are deployed to determine the players' locations.
Interactive storytelling [19, 27] is another research area that has recently aroused widespread interest in the computer graphic and AI community. However a contradiction between interactivity and storytelling requirements lies at the heart of research in this field. The main issue is in fact to allow the user to modify the narrative flow in real-time while preserving the consequences of the user interventions within the main plot.
This paper is structured as following. In the next section, we describe work specifically focused on the generation of multimodal output. We further set forth the description the ideas and background behind our system. We then report on the multimodal generation scheme utilized in our latest prototype for the case of a few conversational situations. Eventually, we conclude with a discussion of current work and future improvements we wish to make.

2 RELATED WORK

Social psychologists assert that more than 65% of the information exchanged during face-to-face communication takes place over the non-verbal channel i.e. through the use of gestures, body posture, facial expression and gaze [25]. In the light of this finding, non-verbal representation is as much important as verbal representation when it comes to developing embodied conversational agents (ECAs) [17, 30].
In recent years, such conversational agents have attracted a considerable amount of attention as a user interface paradigm. Their capability to emulate human–human communication is expected to improve the intuitiveness of a user interface thus increasing both its expressivity and effectiveness. Several animated agent-based systems have been proposed and developed for many different applications, ranging from information presentation, online sales, e-learning and entertainment [13, 15, 17, 23, 24, 28, 32]. Even if significant progress has been made in the graphical appearance and in the quality of synthetic voice, there is no exhaustive framework that overarches all the issues involved in the generation of coordinated and believable multimodal behavior and research still struggles with the realization of paired verbal and non-verbal modes.
Several research groups have addressed the issue of automatic synchronized multimodal generation. The underlying idea is to distribute and synchronize the semantic content of a multimodal utterance across the different output channels at hand. Usually, those systems draw upon a predefined set of elementary animations and, to our knowledge, only in [18] there has been an attempt to create a blend of verbal and non-verbal behaviors based on a theoretical investigation of the relationships between gesture and speech [16]
Based on empirical experiments and evidence, [15] is probably the most cited work that deals with that issue while [35, 36] are two additional contributions that discuss multimodal microplanning architectures for the integration of natural language and gestures for multimodal display. Our work, even if developed independently, resembles a lot and is mostly related to the Improv animation system described in [29]. Improv is a set of tools and techniques for

creating virtual worlds and applications populated with animated agents. Those agents interact, communicate and respond to user input in ways that convey mood and emotion. The system is based on hierarchical sets of scripts that make improvisational animation possible to virtually anyone even without specific knowledge of 3D animation or artificial intelligence.

3 NICE: AN EXPLORATIVE GAME

3.1 Background

We envision a game scenario of a player interacting with embodied fairy-tale characters in a 3D world via spoken dialogue as well as graphical gestures via a mouse-compatible input device in order to solve various problems [6]. The basic idea behind the game scenario is to have the player interact with an agent impersonating the writer Hans Christian Andersen to learn about his life, historical period and fairy tales in a edutaining way. To reinforce the learning experience and make the interaction even more entertaining, the user should also be granted access to a 3D fairy tale world populated by the writer's fairy tale characters. The user can wander about, manipulate objects and collect information useful to solve tasks, which arise while exploring the fairy world, such as e.g. passing a bridge guarded by a witch. For the user to have the impression that she is interacting with distinct, believable agents, each virtual character will have its own proper appearance, voice, actions, and personality.

Our current game system includes a single full-body embodied character, i.e. the writer himself within a graphical 3D representation of his study. There is no user avatar; the user perceives the world through a first-person perspective. She can explore the study and talk to the character, in any order, about any topic within a few knowledge domains, using spontaneous speech and mixed-initiative dialogue (see [11] for a complete system description of the first system prototype). The user can move Andersen about, change the camera view, refer to and talk about objects in the environment, and also point at or gesture to them. Current topic domains are: Andersen's fairy tales, his life, his physical presence in his study, the user, and writer's role as gate-keeper for access to the fairy tale world. Interaction within the fairy tale world and with its characters is also being implemented, yet by another project partner of us [6]. Currently, the two environments – fairy tale world and study - remain separate. There is also a meta domain in order to be able to handle meta-communication during conversation. The character reacts emotionally to user input by displaying emotions and through a combination of synchronized verbal and non-verbal behaviors.

The user is not assigned any specific goal to accomplish. Thus, according to a classification of games and other possible forms of media interactions that has been put forward in [31], our current system is what in jargon is called a 'toy'. Toys are explorative games because they are meant just to be played with and traveled over for adventure, discovery and fun. Flight simulators and Sim City™ [10] are also toys since they have neither objectives nor goals. As will be the case in the final system we envision, adding a goal to our toy (we rather call it 'explorative game') will turn it into a game [31].

3.2 Embodied Character's Output States

Before, during and after conversation, the graphical character keeps on switching among three different possible output states.

While in the communicative function (CF) output state, the character shows his awareness of being spoken to or otherwise addressed by the user, by employing a rather neutral and general set of animations or sequences thereof. The relative neutrality of these behaviors is imposed by the technical limitations which prevent processing of the current user input incrementally in real time. This means that our agent cannot react, while being addressed, to parts of the user input which, given his personality, should otherwise make him react emotionally or cognitively.

A non-communicative action output state (NCA) refers to the situation in which the agent is not engaged in conversation with a user but goes about his normal work and life within the study. This may happen either at system startup if there is no user around, or after the user stops conversation and walks away. In general, while in this state, the character does not produce spoken utterances in terms of full-form sentences. Yet, the system is capable of playing back, e.g., footsteps when he walks, or music sounds when he enacts a dance. Sound files in both wav and mp3 format can be played and only need to be stored in advance to allow for that capability.

The agent is in a Communicative Action (CA) state anytime he takes the turn in producing outputs that serve as a vehicle for the on-going conversation. CA output is the character's actual conversational contribution in the form of verbal realization of one or more speech acts within a single turn, physical actions within the graphical environment,

emotion display, gaze and gestural behaviors. Responses to, and questions for, the user, observations, confirmations and acknowledgements, and meta-communication, such as clarification questions, are produced when in this state.
With regard to state transitions, a default initial NCA state can be followed only by a CF state. This occurs as soon as a user starts addressing the embodied character, initiating an interactive session. The agent himself never starts the conversation for he has no sensor for visually perceiving the environment or detecting a potential interlocutor in other ways. After the user's first conversational contribution, the system keeps on switching back and forth between CF and CA output states to reflect the on-going conversation turn-takings. This may go on for a while, eventually being followed by the NCA state.
From a technical point of view, response planning for non-communicative actions and communicative functions are based on the same scheme while response planning for communicative actions is carried out with a different strategy.

4 SCRIPT-BASED MULTIMODAL OUTPUT GENERATION

In general terms, output generation consists in producing utterances that serve as a vehicle for dialogue. In this context, an utterance refers to both non-verbal and spoken realization of one or more dialogue acts within a single turn is meant. The task of the response generator is to create a coordinated synchronized multimodal output in the form of some meaningful combination of speech and animated character behavior. In addition, the response generator keeps listening to incoming messages from input fusion regarding the selection of on-screen objects by the user. When such an event occurs, Hans Christian Andersen turns towards the selected object. An example is shown in Figure 1 which depicts a few screenshots of the character turning toward the feather pen selected by the user by making a circular trace around it (see the leftmost picture in Figure 1).
To accomplish modality synchronization in the CA output state, the output generation module utilizes both a parameterized semantics generated by the natural language understanding module and a set of utterances retrieved by the character manager. The utterances are analyzed and further split into synchronized text instructions to the speech-synthesizer, and behavioral instructions to the graphical animation engine. The parameterized semantics are used to fill in possible placeholders present in the utterances. A detailed description of the production of communicative actions can be found in [20]; from now on, we focus on the scheme used for the generation of NCA and CF states to elicit a rich interaction experience.

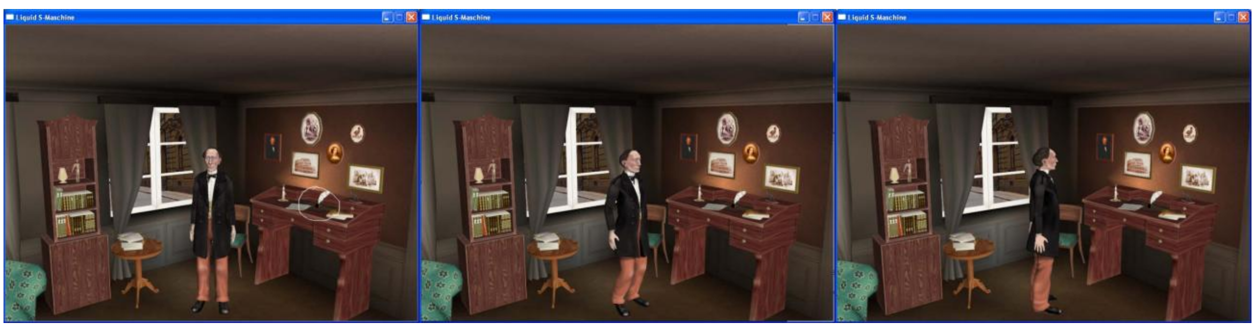

Figure 1: *(from left to right) the character turned to the selected feather pen*

In our implementation, we employ a hierarchical two-tiered approach by supporting the designer in creating the output states through scripts. Lower-level scripts define sets of elementary animations along with their temporal specification, rendering speed and percentage of the animation to play (by default the whole animation i.e. 100% is played) and sets of custom animations. Elementary animations are behaviors that the animation engine can render by a single command. They are sort of building blocks (currently they are 151 of these) and belong to the core of the application as default animations. Custom animations are sequences of such elementary behaviors.
The following is an excerpt from the scripts at the lowest level showing both elementary animations (their names are self-explanatory) and how custom behaviors are generated using a combination of them either in sequence or in parallel:

NOD	LASTS	3000	MSECS
HEADUP	LASTS	3000	MSECS
HEADDOWN	LASTS	2000	MSECS
TILTHEADLEFT	LASTS	1000	MSECS
TILTHEADRIGHT	LASTS	1000	MSECS

SEQUENCE SEQ_STARTMUSIC

SEQ_GOTOCENTER THEN WAIT 4000 MSECS
PLAY SOUND FROM POOL 2
GO TO *Desk*
WAIT 2000 MSECS
SET CAMERA *LookAtActorCamera*
WAIT 2000 MSECS
TURN TO CAMERA FROM POOL 1
WAIT 2000 MSECS

END SEQUENCE

PARALLEL PAR_DANCESTEP1

TRACK WITH PRIORITY 1
ANIMATION HEADUP AFTER 0 MSECS PLAYS 100 PERCENT AT SPEEDRATE 0.5
ANIMATION HEADDOWN AFTER 2000 MSECS PLAYS 100 PERCENT AT SPEEDRATE 0.5

TRACK WITH PRIORITY 2
ANIMATION HOLDOBJECT AFTER 0 MSECS PLAYS 50 PERCENT AT SPEEDRATE 1.0
ANIMATION RESTARMS AFTER 3000 MSECS PLAYS 100 PERCENT AT SPEEDRATE 1.0

END PARALLEL

Elements in *italics* are variable names of objects, camera views, locations and sounds that can be used in the definition of animations. Recursive calls of sequences are allowed. Sequences of elementary and/or custom animations can then be used to define more complex behaviors. To make the character perform a richer variety of behaviors, we make use of placeholders within the sequences for animations and other elements to be selected at run-time. In more detail, one defines several sub-sets (also referred to as pools) of behaviors, objects, locations, cameras and sounds, assigns them a name, and selects them in a non-deterministic way, one each time the placeholder refers to that particular subset. In the example above, for instance, when the SEQ_STARTMUSIC sequence animation is played, different sounds can be played at different times according to the music file chosen at run time from the pool of sounds specified by the command PLAY SOUND FROM POOL 2. The same applies in the choice of the camera, just a few lines later in the behavior description. Each time the behavior is rendered a run-time random function selects the camera view from a sub set of camera definitions.

We also use syntactic rules to define and provide appropriate transitions among animations in order to produce believable and smooth interactive character behavior. An example is the following excerpt from the script that defines the syntactic rules which specifies animation transitions:

SEQ_WALKTOWINDOW CANNOT BE FOLLOWED BY [SEQ_WALKTOWINDOW,ARMSONDESK]

The use of commands such as GO TO *Desk* in SEQ_STARTMUSIC gives the conversational agent own locomotion capabilities and makes the behaviors more life-like and not restricted to a single location. Currently, we still have the

issue of obstacle avoidance. In fact, if the character is sent to a location but an object hinders him in his path to get there, he gets stuck in that state and cannot move further without external help from the user via the use of the keyboard. Our current temporary solution to this problem is to always send the character to the center of the study at the end of a script or after a command that makes the agent move to some location.
At a higher level, more scripts are defined on top of behaviors and their transactions. Here is an excerpt from such a file:

SCRIPT 0

SEQ_WALKTODESK THEN WAIT 4000 MSECS
SEQ_STUDYATDESK THEN WAIT 3000 MSECS
SEQ_LOOKAROUND THEN WAIT 4000 MSECS
SEQ_RESET THEN WAIT 4000 MSECS

Similar to the case of scripts at the lower level, we use syntactic rules to define and provide appropriate transitions among scripts. An example is the rule:

SCRIPT 0 CANNOT BE FOLLOWED BY [0-6,9]

This rule ensures that script number 0, if chosen at run time and after being realized, cannot be followed by scripts 9 and all those from number 0 through 6. In that case, the Character Module, in which the decision about which script to run resides, is constrained in its choice of the next script yet script transition is guaranteed to occur smoothly up front without abrupt movements between the end of a script and the start of the following one.
Figure 2 depicts a couple of screen shots of some non-communicative actions showing the character moving about his study. Also the final screen shot of an animation that makes the agent bend over onto his desk is shown with different camera views as chosen randomly at run-time from different instances of the same animation.

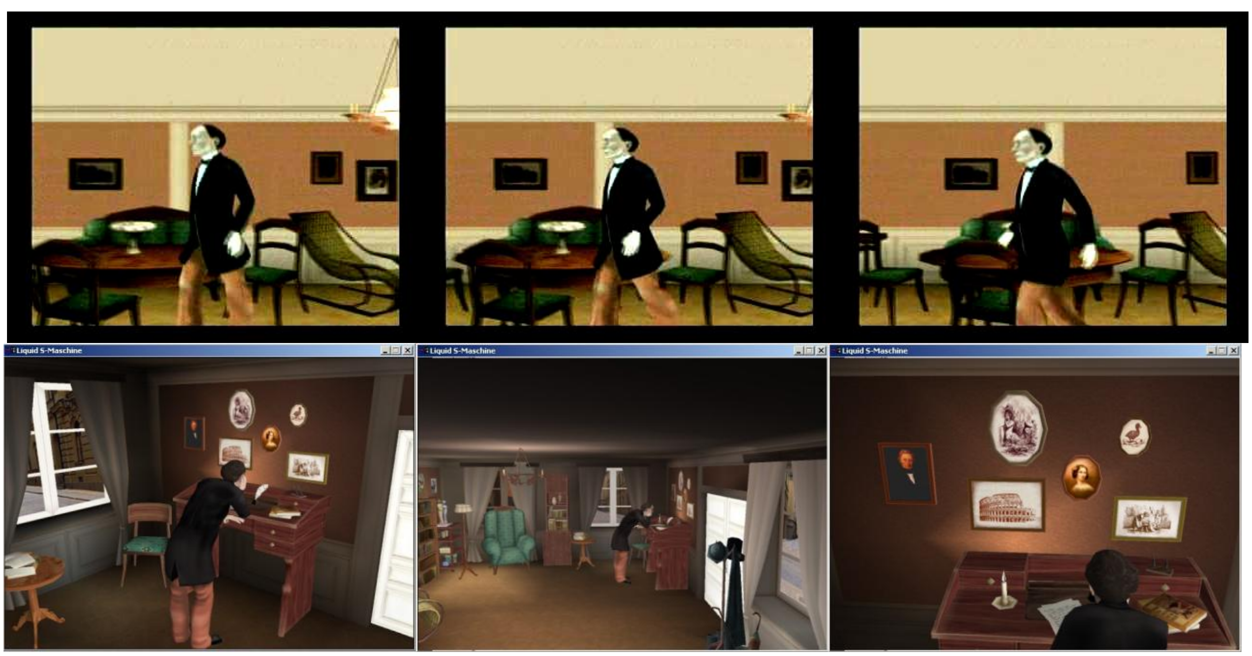

Figure 2: *the conversational agent walking about while in NCA state (top row, left to right) and the agent leaning over his desk from three different camera views (bottom row)*

Since the animation engine can play only elementary animations, the response generator has to break down any user-defined animation of a script into its components and then create a representation string for each of them for the

animation engine to understand. If each of these components is by itself a complex animation, it has to be recursively decomposed, until only primitive animations are exploited to express the original animation.
Rendering of 3D graphical non-verbal behaviors, such as physical action (e.g. pointing at, picking up), display of mood and emotional state, feedback (e.g. nodding) and back-channeling gestures, increase the believability of the fairytale characters. The character can also generate actions on few active objects such us dropping them or picking them up. However, these actions are performed in a primitive manner, as objects are simply put either on his head, if picked up, or back to their original position, if dropped.

4.1 ANIMATION RENDERING

Animating a 3D model or parts of it, means that its position, scale and orientation at different points in time is changed. The animated character is built upon a hierarchy of bones, where each bone represents what we refer to as a frame. The hierarchy of frames together with a textured polygon mesh and skin-weighting information is represented as a skinned mesh. The skinning information specifies the influence a frame has on its mesh. The root frame node contains a transformation matrix relative to the world space. An animation that affects the root node affects the whole scene while an animation that affects a leaf node does not affect any other node. We use the frame hierarchy to give the system the functionality of overloading animations for different parts of the body. Eight priority tracks ensure that only one animation (that with highest priority) out of several ones affecting the same frame is played. The animation system receives network commands and schedules animation events via the scheduler system. These can be started at a particular time as set by the designer or at run time using a common game time common to all modules communicating with the animation engine.

5 DISCUSSION AND CONCLUSIONS

Kids across gender, age groups and culture, like to play video games. Games that support characters behaving like human beings seem to enhance gamers' experience [34]. In consequence, game developers have tried to recreate human behaviors either by using complex body models or by tracking and recording movements to play back during game play. We have presented an approach to multimodal output presentation for an educational game system that we are currently developing. As most of the information exchange in human face-to-face communication takes place via the non-verbal modalities, their coherent realization is as important to the development of our agent as that of verbal output. Rendering of non-verbal behaviors consistently synced up with speech increases its believability. Unfortunately, defining an exhaustive strategy for believable behavior is still an unsolved issue.
While the character is in the CF and NCA states, believable multimodal output generation for our 3D full-body conversational agent is based on a two-tier hierarchy of scripts. We have addressed output generation from a technical perspective rather than from a purely principled one as in [16, 18]. Nevertheless, while in these states, our character is lifelike, reproduces the human physics in detail, and performs non-verbal behaviors in exaggerated manner as this has been proven to convey emotions more efficiently and directly than regular performance, making watching a funny and rewarding experience.
Complex behaviors can be created, combined and sequenced by a generative approach based on a layered composition of primitives. Scripts can be flexibly and easily expanded to include new behaviors at designer will by employing an English-like syntax that makes them easy to create for anyone without any specific knowledge of animation techniques. While fine-control of the motions of single body parts is not possible, the capability of combining primitives along with animation selection at run-time allows creating a large variety of movements, thus compensating for the lack of fine-tuning of single body parts inherent to our animation scheme.
We are currently running a user test of the system, targeted at 10 to 18 year olds recruited from local schools. The most well-known evaluation studies have been directed towards showing the persona effect, referring to the fact that animated agents can have a positive effect on the dimensions of entertainment, motivation, learnability and task difficulty [21]. As concerns response generation, like in [26], on the long run, we are also interested in investigating the likeability of different types of synthetic interface characters.
The auditory realization of tuning voice quality and prosody to tailor the agent's personality is an additional issue that needs to be investigated in the future.

6 ACKNOWLEDGMENTS

We gratefully acknowledge the support for the NICE project [6] from the Human Language Technologies programme, EU contract # IST-2001-35293. We would also like to thank our partner Liquid Media AB for both developing the game engine and collaborating on non-verbal character realization.

References

[1] http://www.becta.org.uk/

[2] http://www.blackandwhite.ea.com/

[3] http://www.microsoft.com/speech/

[4] http://www.microsoft.com/games/empires/multimedia.htm

[5] http://www.gamestudies.org/

[6] http://www.niceproject.com

[7] http://www.idsa.com/pressroom.html

[8] http://www.konami.com/lifeline/

[9] http://www.sierra.com/

[10] http://simcity.ea.com/

[11] Bernsen, N.O., Charfuelàn, M., Corradini, A., Dybkjær, L., Hansen, T., Kiilerich, S., Kolodnytsky, M., Kupkin, D., and Mehta, M. First Prototype of Conversational H.C. Andersen. In: *Proceedings of ACM International Working Conference on Advanced Visual Interfaces (AVI '94)*, Gallipoli (Italy), 458-461, 2004

[12] Björk, S., Falk, J., Hansson, R., and Ljungstrand, P. Pirates! Using the Physical World as Game Board. In: *Proceedings of the Second International Conference on Entertainment Computing***,** May 8-10, Pittsburgh (PA), 2003

[13] Bosseler, A., and Massaro, D.W. Development and evaluation of a computer-animated tutor for vocabulary and language learning in children with autism, *Journal of autism and developmental disorders*, 33(6):653-72, December 2003

[14] Callois, R. *Man, Play, and Games*. New York: The Free Press, 1961

[15] Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjálmsson, H.H., and Yan, H. Embodiment in conversational interfaces: Rea, *Proceedings of CHI*, Pittsburgh (PA), 520-527, 1999

[16] Cassell, J., and Stone, M. Living Hand to Mouth: Psychological Theories about Speech and Gesture in Interactive Dialogue Systems, In: *Proceedings of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 34-42, 1999

[17] Cassell, J., Sullivan, J., Prevost, S., and Churchill, E. (eds), *Embodied Conversational Agents*, MIT Press, 2000

[18] Cassell, J., Vilhjálmsson, H.H., and Bickmore, T. BEAT: the Behavior Expression Animation Toolkit, In: *Proceedings of SIGGRAPH*, 477-486, 2001

[19] Cavazza, M., Charles, F., and Mead, S.J. *Character-based Interactive Storytelling.* IEEE Intelligent Systems, special issue on AI in Interactive Entertainment, 2002, 17-24

[20] Corradini, A., Mehta, M., Bernsen, N.O., and Charfuelàn, M. Animating an Interactive Conversational Character for an Educational Game System. In: *Proceedings of the International Conference on Intelligent User Interfaces (IUI '05)*, San Diego (CA), 183-190, 2005

[21] Dehn, D.M., and van Muelken, S. The impact of animated interface agents: a review of empirical research, *International Journal of Human-Computer Studies*, Volume 52, Issue 1, 1-22, 2000

[22] Huizinga, J. *Homo Ludens: A Study of the Play-Element in Culture*, Beacon Press, 1971

[23] Johnson, W.L., Shaw, E., Marshall, A., and LaBore, C. Evolution of user interaction: the case of agent Adele, *Proceedings of the International Conference on Intelligent User Interfaces (IUI '03)*, 93-100, Miami (FL), 2003

[24] Johnson, W.L., Rickel, J.W. and Lester, J.C. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments, *International Journal of Artificial Intelligence in Education* 11, 47-78, 2000

[25] Knapp, M.L. *Non-verbal Communication in Human Interaction*, second edition, Holt, Rinehart and Winston Inc., New York, 1978

[26] McBreen, H., Shade, P., Jack, M. and Wyard, P. Experimental assessment of the effectiveness of synthetic personae for multi-modal e-retail applications. In: *Proceedings 4th ACM International Conference on Autonomous Agents*, 39-45, New York City (NY), 2000

[27] Mateas, M., and Stern, A. *A Behavior Language for Story-Based Believable Agents*. IEEE Intelligent Systems, special issue on AI in Interactive Entertainment, 2002, 39-47

[28] Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F., and Poggi, I. Embodied Contextual Agent in Information Delivering Application, In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 758-765, 2000

[29] Perlin, K., and Goldberg, A., Improv: A System for Scripting Interactive Actors in Virtual Worlds, Computer Graphics, 29(3):1-11, 1996

[30] Picard, R. *Affective Computing*, MIT Press, 1997

[31] Prensky, M. *Digital Game-Based Learning*. McGraw Hill, 2001

[32] Prendinger, H., and Ishizuka, M. (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*, Springer Verlag, 2004

[33] Starner, T., Leibe, B., Singletary, B., and Pair, J. MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game, In: *Proceedings of Conference in Intelligent User Interfaces*, 256-259, 2000

[34] Sweetser, P., Johnson, D., Sweetser, J., and Wiles, J. Creating Engaging Artificial Characters for Games. In: *Proceedings of the second International Conference on Entertainment computing*, Pittsburgh (PA), 2003

[35] Tepper, P., Kopp, S.,and Cassell, J. Content in Context: Generating Language and Iconic Gesture without a Gestionary, In: *Proceedings of the Workshop on Balanced Perception and Action in ECAs* at AAMAS '04, New York City (NY), 2004

[36] Theune, M., Heylen, D., and Nijholt, A. Embodied Language Generation, In: Stock, O., and Zancanaro, M. (eds.) *Multimodal Intelligent Information Presentation*, Kluwer Publisher, 47-70, 2004