Animating an Interactive Conversational Character for an Educational Game System

Andrea Corradini, Manish Mehta, Niels-Ole Bernsen, Marcela Charfuelan
Natural Interactive Systems Laboratory (NISLab)
University of Southern Denmark
DK-5230 Odense M, Denmark
+45 6550 3698
{andrea,manish,nob,marcela}@nis.sdu.dk

ABSTRACT
Within the framework of the project NICE (Natural Interactive Communication for Edutainment) [2], we have been developing an educational and entertaining computer game that allows children and teenagers to interact with a conversational character impersonating the fairy tale writer H.C. Andersen (HCA). The rationale behind our system is to make kids learn about HCA's life, fairy tales and historical period while playing and having fun. We report on the character's generation and realization of both verbal and 3D graphical non-verbal output behaviors, such as speech, body gestures and facial expressions. This conveys the impression of a human-like agent with relevant domain knowledge, and distinct personality. With the educational goal in the foreground, coherent and synchronized output presentation becomes mandatory, as any inconsistency may undermine the user's learning process rather than reinforcing it.

Categories and Subject Descriptors
H.5 [**Information Interfaces and Presentation (e.g., HCI)**]: Multimedia Information Systems – *animations;* User Interfaces – *natural language, user-centered design.*

K.3 [**Computers and Education**]: General

General Terms
Design, Human Factors.

Keywords
Edutainment, embodied conversational agent, multimodal output, user interface.

1. INTRODUCTION
To convey information to one another, humans use a broad set of modalities, such as speech, body posture, gesture, facial expression, text, object-manipulating action, drawings, and even clothing. People elaborate on these signals by exploiting their multiple sensory systems including, e.g., vision, olfaction, haptic, and audition to create a unified representation of the input signal, which is then contextually interpreted.

In human-computer interaction (HCI), research in user interfaces has mainly been done to provide machines with the ability to deal with a strongly restricted set of input/output modalities. Thus, most of these computer interfaces are still based on the two-decade old WIMP (Windows, Icons, Menus, Pointing devices) paradigm supporting graphical user interfaces (GUI) and conventional input devices. One of their major limitations, i.e., the requirement of sequential and unambiguous input, is also their greatest strength since it allows building very robust and feasible interfaces for standard applications in office settings. However, GUIs do not scale up to an arbitrary increase of interface complexity.

Multimodal interfaces relax GUI constraints by allowing ambiguous, asynchronous, and inaccurate input, and by massively increasing input/output expressiveness. They can deal with more complex tasks that give the user the flexibility to employ the most appropriate communication modalities for the immediate circumstances. On the other hand, they require more sophisticated techniques for the analysis and fusion of natural input modalities including speech and language, gesture, and gaze [19, 20, 40]. Multimodal interfaces do not focus on input only. They also address the issues related to the capacity of a computing system to convey meaning to the user automatically using different output modalities. So far, though, only little attention has been paid to the coordination of output modalities. Hence, the development of advanced output modules will be one of the main future directions for this kind of interfaces [33] since machines able to integrate input modalities and simultaneously generate coordinated multimodal output are better candidates for transferring the naturalness of human-human interaction into HCI systems that emulate the way humans communicate with one another.

To set the scene, in the next section we take a brief look at the ties between play and education. We then review related work and describe our system architecture with emphasis on the module for the realization of multimodal behaviors, its key features and the

IUI'05, January 9-12, 2005, San Diego, California, USA.

capabilities it offers. Eventually, we conclude with a discussion of current work and future improvements we plan to make.

2. GAMES AND EDUCATION

Computer and video games are among the most entertaining and engaging pastimes and forms of fun for kids across gender, age groups and culture [1]. By deploying the increasing processing power and memory currently available in PCs and consoles, computer games are becoming more and more complex and diverse. Their scenarios are usually characterized by high-quality graphics and sound to offer immersive 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.

Researchers, philosophers, and social scientists relate play to sociological concerns and biological functions that have to do specifically with learning [14, 23, 38]. Besides providing pleasure and challenge, games may motivate the user to increase her involvement and contribute towards educational objectives [38]. The belief that, in addition to traditional play activities, humans are captivated by other forms of learning through playing, is supported by some TV shows, such as *Sesame Street* and *the Smurfs,* that have entertained and educated entire generations of children [13].

3. RELATED WORK

Animated interface agents are now being used in a wide range of application areas, including personal assistants, e-commerce, entertainment, and e-learning environments. A growing research community is working on the development of embodied conversational agents (ECAs) that coordinate synthesized speech and non-verbal behaviors [17].

Rea [15] plays the role of a real estate salesperson. She uses gaze, head movements and facial expressions for dialogue functions such as turn-taking, emphasis and greetings. Verbal and non-verbal communication aspects of Rea are limited to task-oriented dialogue in which the agent interacts with users to determine their needs, shows them around virtual properties, and attempts to sell them a house. The dialogue is entirely Rea-initiated. Baldi [30] is a 3D talking head with accurate articulatory movements and facial expressions. It is used to create a realistic conversational agent for language training, in particular for autistic children and the hearing-impaired. Greta [35] responds to user queries by speaking and exhibiting gaze, head movements, and facial expression. She can express emotions consistently with the context, and she has her own personality and social role. CrossTalk [4] is a project that involves two separate screens where different animated agents have cross-screen conversations amongst themselves. It explores a new variation of information presentation in public spaces, such as a booth at a trade fair or a product information kiosk in a department store. CrossTalk involves a virtual hostess who in turn introduces two additional agents that engage in a car sales dialogue. Smiley [32] is an animated agent used in a simple courseware to provide feedback to the students based on their progress in training. Adele [24] is another animated agent which has been used in a case-based clinical diagnosis application where students are presented with materials on a particular medical condition and are then given a series of cases that they are expected to work through. Adele highlights interesting aspects of the case, and monitors and gives feedback as the student works through a case. Steve [25] is designed to interact with students in networked immersive virtual environments. Steve has been used in training people to operate engines aboard US Navy surface ships.

The issue of creating coordinated multimodal output has been addressed for multimedia presentation as well. The PPP Persona [9] was an early attempt at using a 2D cartoon-like character for displaying and commenting on graphical and textural output. In [34] a web-site agent has been developed to assist the user with navigating across the visited web-site and receive recommendations for related documents. W3C's Synchronized Multimedia Activity [6] has focused on the design of a new language for choreographing multimedia presentations in which audio, video, text, and graphics are combined in real-time. It enables authors to specify what should be presented when, enabling them to control the precise time that a sentence is spoken and make it coincide with the display of a given image appearing on the screen.

All the works we have briefly reviewed differ from our HCA system by being limited to task-oriented scenarios, except Rea [15], which can also simulate small talk largely by ignoring the content of the user's replies to her sentences. While all these ECAs either entirely initiate the dialogue or are not capable of initiating output on their own, our agent is able to deal with both situations. In addition, only Rea is a full-body 3D agent able to generate spoken and non-verbal output behavior in response to unimodal spoken input from the user. Rea does actually also passively sense user movements by means of two cameras that track hands and head. However, it seems that (few) movements or poses are considered in isolation for regulating the flow of conversation, notably to determine cues associated with turn-taking behaviors, rather than being combined with speech to extract a common semantics from the two modalities. Arguably, moreover, web-site agent presentations ought to be called multimedia output presentations rather than multimodal output generated behaviors because the modalities are not generated but only played back from a prerecorded set of behaviors.

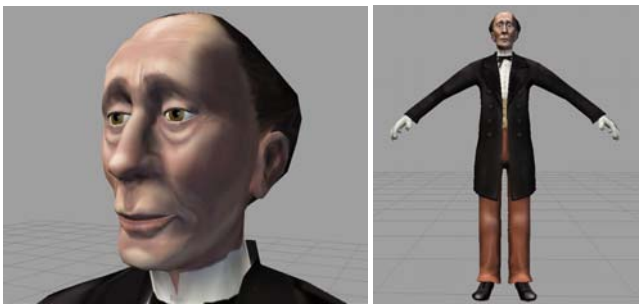

Figure 1. ***(left to right)*** **HCA's face details and full body**

4. NICE: AN EXPLORATORY GAME

4.1 Background

We envision a game scenario of a player interacting with embodied fairy tale characters in a 3D world, using spoken conversation as well as 2D gesture. The basic idea behind the game scenario is to have the player interact with HCA to learn about the writer's life, historical period and fairy tales in an entertaining way. To reinforce the learning experience and make

the interaction even more challenging, the user is also granted access to a 3D fairy tale world populated by some of HCA's fairy tale characters. The user can wander about, manipulate objects, and collect information useful to solve plots which arise while exploring the fairy world, such as finding out how to pass a bridge guarded by a witch. For the user to have the impression of interacting with individual, believable agents, each virtual character will have its own proper appearance, voice, actions, and personality.

In our current real-time implementation, the game scenario includes a single, fully embodied character impersonated by HCA (Figure 1). There is no visible user avatar, as the user perceives the world around him in a first-person perspective. She can explore HCA's study and talk to him, in any order, about any topic within HCA's knowledge domains, using spontaneous speech and mixed-initiative dialogue. The user can control the locomotion of the agent, change the camera view, refer to and talk about objects in the study, and also point at or gesture to them. Typical input gestures are markers like, e.g., lines, points, and circles, entered at will via a mouse-compatible input device or using a touch-sensitive screen. HCA's domains of discourse are: his fairy tales, his life, his physical presence in the study, his role as gate-keeper for access to the fairy tale world, the user, and the meta domain of solving problems of meta-communication during speech/gesture conversation. Interaction within the fairy tale world and with its characters is currently being implemented. Thus, the two environments – the fairy tale world and HCA's study – still remain separate.

Since the user is not assigned any specific task to accomplish, according to a classification of games and other possible forms of media interactions that has been put forward in [38], our current system is what in technical jargon is called a *toy*. Toys are explorative games that have neither objectives nor goals. They are meant just to be played with and/or traveled over for adventure, discovery and fun. Flight simulators and Sim City™ [3] are examples of toys. As will be the case in the final system we envision, adding a goal to our toy will turn it into a game.

4.2 General Architecture Overview

Our current real-time prototype relies on a distributed agent architecture [5]. Software agents, such as for natural language understanding (NLU), the animation module, etc., communicate with one another via TCP/IP by means of a broker whose task is to route messages, results, and error codes among modules in a common XML format.

In terms of information flow, a speech recognizer and a gesture recognizer both send n-best hypothesis lists to the NLU module and the gesture interpreter, respectively. The gesture interpreter consults the animation module to figure out which on-screen objects, if any, the user has referred to while drawing a gesture. Output from the natural language module, consisting of a domains/topics/semantics frame, and n-best output from the gesture interpreter, are then forwarded to the input fusion module. At present, this module simply forwards the two top-ranked inputs to the character module (CM) that is responsible for response planning, updating HCA emotional state, and keeping track of the conversation history. Eventually, the response generator (RG), using information from the CM, including spoken output template references, non-verbal behavior references, user input values, and a representation of HCA's current emotional state, generates text-to-speech and graphical animation output, including their timing, for final synchronization by the animation module [10, 11].

5. MULTIMODAL OUTPUT GENERATION

5.1 Animations and 3D Graphical System

Our animated character is built upon a hierarchy of bones that we refer to as a frame. The frame, together with a textured polygon mesh and skin weighting information, is represented as a skinned mesh (Figure 2). The skin weighting information specifies the influence the 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 to give the system the functionality of overloading animations for different body parts.

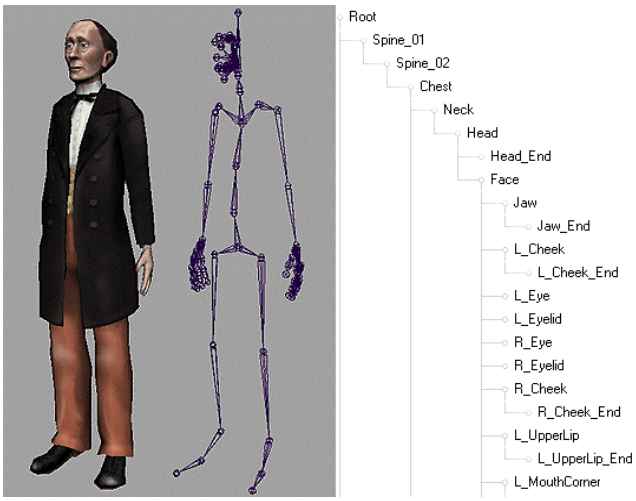

Figure 2. Skinned mesh of the HCA character

The animation system receives network commands and schedules animation events via the scheduler system. A valid command to concurrently carry out sequences of animations while synthesizing a sound file is an XML string whose syntax looks like:

<play>
<sound> SOUND </sound>
<animList Track= P_1> T_{11},A_{11}; .. </animList>
<animList Track=P_2> T_{21},A_{21}; .. </animList>
</play>

The tag <sound> is utilized to play back the audio file SOUND either as positioned or global sound. The items <animList Track=P_j> contain a list (or track) of animations A_{11},.. to be played in sequence in accordance with their start times T_{11},.. relative to a common time. To resolve conflicts that may occur while animations are rendered in parallel, each track is assigned a priority value P_j. If at any given time more animations affect, in different ways, the same node within the frame, the animation

within the track with higher priority will prevail, i.e., be displayed. In general, an animation can be played only if an existing file that contains the specification of the actual behavior has been loaded into memory by the rendering engine upon start-up. We refer to these animations as elementary animations or primitives.

5.2 The Character Module

Any time HCA has to produce a response or initiate a new conversational turn, the CM selects a contextually appropriate output in accordance with the following factors: the semantic categories provided by the NLU, the conversational history, the emotional state, and a set of logical preconditions and output states. Logical preconditions typically occur within short, predefined, guided dialogue (mini-dialogue) that allows HCA to carry out in-depth conversation on key topics. The output is selected either from the knowledge base in form of canned templates that represent both verbal and uninstantiated non-verbal behaviors in a compact way or fetched from the Internet.

5.2.1 Conversation History and Domains

From the CM's point of view, a domain is a categorization of a set of semantic classes provided by the NLU. Currently, HCA can keep up a dialogue about the following domains: HCA's study, physical presence, life, and fairy tales, user and meta-communication. Topics are intra-domain sub-categorizations.

The CM maintains a conversation history that keeps track of information about the individual input and output turns, domains and topics talked about, number of consecutive turns involving meta-communication, and mini-dialogue status (such as start, ongoing, end). A conversation history is maintained both to prevent HCA from repeating himself during conversation and to help HCA select a new conversational domain. A new domain is selected when HCA either runs out of sentences/topics within the current one or the current domain is declared covered, i.e., a certain number of conversational exchanges within it has occurred in a row. HCA does not address topics of covered domains anymore unless they are explicitly reintroduced by the user. Thus, replies to the user input are repeated only if the user provides the same input several times.

5.2.2 Modeling Emotions

HCA has a fairly simple emotional state-space model. The default state can shift along three dimensions (happiness, anger and sadness) to reflect his emotional reaction to the user's input [11] as the dialogue proceeds. Emotion decrements/increments are attached to the output templates stored in the knowledge base. Whenever a template which carries an emotion increment, is retrieved, an update of the emotional state takes place. The strength of HCA's non-default emotions decreases over time [37]. If the user input does not elicit any emotion increments, HCA's emotional state nudges back towards the default (friendly) state.

5.2.3 Response Planning and Output States

The CM keeps on switching back and forth among a set of three output states determined by the conversational flow.

A non-communicative action output state refers to the situation in which HCA is idle, i.e., not engaged in conversation with a user. Such a state is entered anytime no input is sensed over a certain period of time. While in this state, HCA goes about his normal work in his study. He does not freeze, remaining inactive until the next user comes along and engages in conversation, for that would look extremely artificial and not at all life-like.

While in the communicative function [17] output state, the CM produces behaviors that make the user believe that HCA is aware of being spoken to. However, due to the technical limitations in today's conversational software agents, the CM cannot semantically process user input incrementally in real time. Rather, it has to wait for the user to stop talking and/or gesturing before starting to process the input. This implies that HCA cannot react, whilst being addressed, to those parts of the user input which should otherwise make him react emotionally or cognitively. To have HCA behave as natural as possible while the CM is in that state, we make him preserve a rather neutral face while, in a non-deterministic way, performing few non-verbal behaviors, such as, e.g., looking at the user, raising the eyebrows, tilting the head, or nodding, that deliver the user the illusion that HCA is attending to what he/she is saying.

Finally, the CM is in a communicative action output state anytime it is HCA's turn in producing conversational output. When in that state, and whenever possible, the CM looks for a contextually appropriate in-domain response. Otherwise, either meta-communication is initiated, such as when the input was not understood, or the CM proposes a way to continue the conversation, which may or may not relate to the topic and/or domain as brought up by the user. Whichever be the case, the CM forwards to the RG either a reference to a linguistic and behavioral template retrieved from the knowledge base or a pointer to/within a mini-dialogue or a string retrieved from the Internet.

5.3 Synchronized Output Realization

5.3.1 Template-based Output

When dialogue occurs within one of the domains listed in subsection 5.2.1, the CM looks for the HCA dialogue contribution within a knowledge base that stores many predefined sentences along with encoded non-verbal behaviors, semantic classes and the system's domains ontology. Numerous, ever-expanding, canned templates guarantee broad domain coverage but also require manual maintenance and have a limited variability by design. Each template is a compact representation of a predefined spoken output with embedded start and end tags for non-verbal behaviors and placeholders to textural values to be filled in at run-time. The following is an example of behavioral template:

> *Now, tell me [g0] your [/g0] {EMOTION ADJ_2} opinion about {FAIRYTALE}*

Here, elements within square brackets starting with numbered *g* letters represent onset and offset of non-specified non-verbal parts of the template. Elements within curly parentheses, like *FAIRYTALE,* are placeholders for text-to-speech (TTS) values to be filled in using input value information. The other elements within curly parentheses, starting with the string *EMOTION*, are TTS values as well but these are tied to emotional values. In the present example, *ADJ_2* indicates a set of emotional value/text pairs from which the verbal realization for the appropriate text has to be retrieved. Both TTS variable values and non-verbal behavioral elements are initially uninstantiated. The binding of

non-verbal behavior to gesture and TTS variables to text occurs at run-time rather than being hard-coded, enabling a sentence to be synthesized at different times with different accompanying non-verbal elements and/or words.

The pair of tags that marks start and end of any non-verbal behavioral element supplies implicit timing information for speech and gesture during rendering. In the behavioral template above, tags *[g0]* and *[/g0]* indicate that an animation may co-occur with uttering the spoken text '*your'* around which they are wrapped. A certain gesture is selected for insertion in place of *g0* depending on the semantic class(es) of the text surrounded by the placeholders. Tables that map semantic categories onto non-verbal behaviors are maintained. Let us assume that a *POINT* animation is selected to expand the non-verbal behavior *g0*, while the textural placeholders *EMOTION ADJ_2* and *FAIRYTALE* are expanded to *valuable* and *the Princess and the Pea,* respectively. The behavioral template is then converted into the surface language string:

Now, tell me [POINT] your [/POINT] valuable opinion about the Princess and the Pea

We have implemented two strategies to deal with such a surface representation. In a first approach, similarly to [21], the RG replaces non-verbal behavior references with bookmarks that can be dealt with by a text-to-speech component. Then, the entire string containing the TTS bookmarks is sent to the TTS, which synthesizes the verbal output. Anytime a bookmark is encountered, the TTS fires an event and calls on the response generator to create the XML string representation of the corresponding animation. The XML string is sent immediately to the graphical 3D engine for rendering. In a second approach, we first parse the surface string for the TTS module to create wav files of text enclosed within animation bookmarks and determine its temporal duration. During parsing, the surface string is broken down into sequential segments of either audio-only segments, animation-only segments, or parallel audio and video segments. Three XML strings would be generated when parsing the surface string of our previous example:

```
1) <play> <sound> SOUND_1 </sound> </play>
2) <play>
        <sound> SOUND_2 </sound>
        <animList Track=0>
                0, POINT;
        </animList>
  </play>
3) <play> <sound> SOUND_3 </sound> </play>
```

Here, *SOUND_1* contains the synthesized text *now tell me, SOUND_2* the text *your*, and eventually the verbal synthesis for *valuable opinion about the Princess and the Pea* is stored into *SOUND_3*. The animation *POINT* is stretched over a time period equivalent to the duration of the sound file *SOUND_2*. Once all XML segments are created, they are sequentially sent to the graphical animation engine that automatically coordinates playback of sound and non-verbal behavior rendering for each of them. This approach is suitable for short behavioral templates because it requires the data to be analyzed twice: first parsing the template to create single wav files, then go through it again to break it down in single segments to send to the animation engine.

In our current implementation we use the first approach for two main reasons. First, we have several long templates that we will be breaking down as sequences of shorter ones. Also, we are having technical problems in fine-tuning the duration of single animations for each of them to last exactly as long as the sound files they play along with independently of the machine that runs the application.

In either approach, in addition to elementary animations, more complex non-verbal behaviors can be created, combined, sequenced, assigned a name, and stored by the RG. To that extent, we have employed a generative approach based on a layered composition of primitives, partially borrowed from [36]. This process consists of the following hierarchically arranged functional elements. First, complex behaviors are designed as sequences of primitives. At the next level, such sequences are assigned priorities and composed to run concurrently. Finally, at the top-most level, scripts, i.e., sequences of complex animations are defined to synthesize long, continuous, non-repetitive behaviors and to allow for smooth transitions between them. Since transitions from certain behaviors into some others may sometimes result in awkward motions, we use rules to either allow or prohibit transitions. In addition, to have a high variability in our scripts and thus non-deterministic synthesis of motions, we allow behaviors within a script to be chosen at run-time from subsets defined by the designer. For example, let's say we have a subset *TURN* made up of {*TURN_RIGHT, TURN_LEFT*} behaviors and a script *THINK_THRU* defined as sequence of [*SCRATCH_HEAD*; *WAIT 3, RANDOM FROM TURN; WAIT 2, FROWNED_EYES; WAIT 3*]. Anytime the script *THINK_THRU* is executed, it sequentially makes HCA scratch his head, wait for 3 seconds, choose randomly an animation from subset *TURN*, wait for 2 seconds, frown and eventually wait for 3 more seconds. Animations within subsets can also be weighted to bias the random selection. Sets of rules over the scripts/behaviors ensure smooth transitions between scripts/behaviors.

Since the rendering engine can play only elementary animations, the RG has to break any user-defined animation down into its primitive components and create XML representation strings for each of them. *THINK_THRU* would thus have to be split into its three defined components at run-time. In turn, these latter have to be recursively decomposed until only primitives are used to express the parent animation *THINK_THRU*. Sequentiality, parallelism, and partial overlapping of existing animations to create new behaviors can be tuned by setting appropriate values for the temporal items in the XML representation.

Currently, we store some 300 templates, many of which are no-variable stories to be told by HCA, and 110 different non-verbal primitives. Templates have been designed by hand and, similarly to non-verbal behaviors, were partly inspired by analysis of data from recordings of an actor impersonating HCA and interacting with kids in a children's theatre class in the fairy tale writer's hometown Odense, Denmark.

5.3.2 Output Retrieval from the Internet

When the processing of the user input results in certain domains which are potentially relevant for the scenario of our system but

for which we have not created any output, we exploit the web as our external knowledge base. Examples of such domains are: computer games, traditional games, jokes, and, in principle, any domain related to game activities that are likely to be either a fashionable topic over a limited period of time or evolve over time and thus need periodic update. For example, we noticed that kids often ask about Harry Potter. While we expect this to continue for some time, we know neither how long nor the names of places, protagonists, among many others that may come up in a sequel of Harry Potter's adventures.

It is to be noticed that anytime the user input is classified into one of those domains, HCA will not be able to continue a contextual conversational exchange beyond the sentence he selects as reply. In other words, the current output is considered in isolation because it is meant to just provide the user with a meaningful reply in a potentially interesting domain not covered by stored templates. To that extent, a quick and concise output is searched for from the web using three freely available open-domain Question Answering (QA) systems: AnswerBus [41], Start [27], and AskJeeves [8]. After four seconds we time-out the query to these QA systems and rank the best results according to few empirical rules. For example, AskJeeves offers a way to signal when a sentence it extracts has high confidence. Empirically, we have noticed that in such cases AskJeeves's result overrules those from the others QA systems we use.

Once a sentence is selected, we remove control/graphical characters and the contextual information peculiar to the QA systems. This process yields a plain string that can be played directly by the TTS but lacks any non-verbal cue. In a second step, we process this string in search of segments that can be semantically categorized. Again, by using pre-stored mappings from semantic categories onto non-verbal behaviors, a suitable motion is then attached to these segments. For instance, in response to '*do you know what high jump is?'* HCA retrieves from the web the sentence '*a competition that involves jumping as high as possible over a horizontal high bar'* and attaches a jumping behavior to the word *jumping* based on its semantic category *jump*.

5.3.3 Mini-dialogues

Mini-dialogues are implemented as state machines [13], thus different conversational paths are possible within a mini-dialogue, depending on the actual user input. Yet, mini-dialogues cannot be exited, i.e., once entered, and even if the user input does not match semantically the current state, HCA keeps on selecting his conversational contributions until the state machine is traversed through to (one of) its end state(s). Each state is assigned a predefined set of output templates. The selection of the current one is carried out according to the conversation history to avoid repetitions and to allow for variability in the sentences.

5.3.4 Other Output Cases

The TTS fires 21 viseme events. Regardless of the state, we always map them onto the 12 mouth articulations standardized by the Disney animators [26], which we have reproduced as primitives to speed up rendering. In case of parallel rendering, visemes are always allotted the track with highest priority. When HCA is in either communicative function state or non-communicative action state, the CM asynchronously and randomly picks up user-defined scripts. These scripts differ both in temporal length and content from one state to another. In the non-communicative action state they tend to be longer in time and have more variability in their visual realization. We use the same scripting approach as for template-based output because it ensures high flexibility and non-deterministic behaviors. At run-time, the script to be realized is randomly selected and broken down into its elementary behaviors. The primitives are sent to the RG that creates the XML representation to be played by the rendering engine. Most of these behaviors are non-verbal; only a few have a non-speech audio component, e.g., for playing footsteps when HCA moves about in his study.

Regardless of the current output state, the RG keeps on sending blinks at irregular intervals to the animation engine. Even if, at first, blinks may seem like the most mechanical side to agent animation, as noted in [26], they indeed largely contribute to keep a character alive.

5.4 Example of Multimodal Conversation

Table 1 shows an excerpt from a conversation between a child user and HCA that has been recorded during a recent user study we run to test the entire system [12]. As the user never makes use of 2D gestures, technically, the input is unimodal. HCA's responses are mostly multimodal with facial expression, spoken utterance and gestural components.

At turn #9, HCA starts meta-communication due to the absence of (gestural and spoken) user input over a certain period of time. The gestures occurring in that dialogue can vary, depending on the mapping from the semantic categories onto non-verbal behaviors. The CM may also decide to leave some of them uninstantiated. Thus, e.g., at turn #8 less than three gestures are attached to the output.

Table 1. Excerpt from conversation between HCA and a child

turn #7

Child's spoken input: *ehm how old are you now?*

HCA output: *I am 55 years old. [g0] And I am in a good shape [/g0] occasionally I ride on horses and go for long walks and [g1] when I travel [/g1] I even sometimes hike*

Animations selected:

g0 = ARMS_CROOKED

g1 = JUMP

turn #8

Child's spoken input: *okay*

HCA output: [g0][/g0][g1][/g1][g2][/g2] *my shoe size is 48 [g3][/g3]*

Animations selected:

g0 = RAISE_EYEBROWS

g1 = LAUGHING

g2 = LOOK_AT_DOWN

g3 = SMILE

turn #9

Child's spoken input: NONE

HCA output: s*ilence is gold but silver is very beautiful*

Animations selected: NONE

turn #10

Child's spoken input: *what are you doing now?*

HCA output: *I am doing many different things in my study. I am [g0] writing, [/g0] [g1] thinking, [/g1] [g2] reading, [/g2] singing, dancing, and so on.*

Animations selected:

g0 = LEFTARM_CIRCULAR_MOVE

g1 = RAISE_EYEBROWS

g2 = FURROWED_EYES

6. DISCUSSION AND CONCLUSIONS

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 ECA development as that of verbal output [17, 37]. Rendering of non-verbal behaviors consistently synced up with speech increases ECA believability. Unfortunately, defining an exhaustive representation that encapsulates all of the attributes necessary for believable behavior is still an unsolved issue.

We have proposed an approach to create believable multimodal output for a 3D full-body conversational agent impersonating fairy tale writer H.C. Andersen that accounts for an entertaining and educational interaction.

One problem with the use of templates in our approach is that, while a gesture has not yet been chosen during design, its timing is already defined relative to speech due to the start and end tags. We wish to point out that, due to the technical complexity and real-time requirements on our system, we have addressed output generation from a technical perspective rather than from a purely principled one as in, e.g., [16, 18]. Nevertheless, 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 (in fact, caricaturists take advantage of it), making interaction a fun experience.

Several XML-based languages have been proposed for specifying human communicative behavior [29] but so far none of these can fully capture the non-verbal information conveyed by humans. Most approaches use own complex mark-up languages to define in great detail movements of single body parts and joints [7]. We use instead a high-level XML formalism to describe the overt form of non-verbal communication by utilizing a simple parameterized library of a few handcrafted behaviors. This library can be flexibly and easily expanded to include new elements for modeling additional behaviors, emotions and moods. The main limitation of our approach is the inherent impossibility to fine-control the motions of single body parts. However, with the capability to combine primitives we can cover a large variety of movements, thus reaching a compromise between the number of predefined behaviors and the complexity of generating others from them. The avoidance of inconsistencies that may deceive and mislead children during multimodal realization helps reaching the educational goal of the system. Incoherent output realization is, indeed, counterproductive as it undermines the child's learning process rather than reinforcing it. Embodiment enhances entertainment and effective user engagement. A recent user test of the system [12], which we carried out with eighteen 10 to 18 year olds recruited from local schools, seems to support these claims.

The system is undergoing continuous improvement and new non-verbal behaviors are constantly being added. Apart from them we are working also on the auditory realization to tune voice quality and prosody to tailor HCA personality.

7. ACKNOWLEDGMENTS

We gratefully acknowledge the support from the Human Language Technologies programme, EU contract # IST-2001-35293. We also wish to thank Liquid Media AB, Stockholm, Sweden, for creating the 3D animation software and Abhishek Kaushik at Oracle India Ltd., Hyderabad, India, for programming support in the agent that retrieves outputs from the Internet.

8. REFERENCES

1. http://www.gamestudies.org/
2. http://www.niceproject.com
3. http://simcity.ea.com/
4. http://www.dfki.de/crosstalk/
5. http://www.speech.kth.se/broker
6. http://www.w3.org/AudioVideo/Activity.html
7. http://www.vhml.org/workshops/AAMAS/papers.html
8. http://www.ask.com
9. André, E., et al. The PPP persona: a multipurpose animated presentation agent, *Proceedings of the ACM International Conference on* A*dvanced Visual Interfaces (AVI)*, 245-247, 1996
10. Bernsen, N.O., et al. First Prototype of Conversational H.C. Andersen, *Proceedings of the ACM International Conference on* A*dvanced Visual Interfaces (AVI)*, 458-461, 2004
11. Bernsen, N.O., and Dybkjær, L. Domain-Oriented Conversation with H.C. Andersen*. Proceedings of the Workshop on Affective Dialogue Systems (ADS*), Lecture Notes in Artificial Intelligence 3086, Springer Verlag, 305-308, 2004
12. Bernsen, N.O. and Dybkjær, L. Evaluation of spoken multimodal Conversation. In: *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI)*, Penn State University (PA), USA, 38-45, 2004
13. Bruce, L., and Bodnar, C. Identity, TV and Papa Smurf: As children of the 1980s, we have been shaped by what we watched, Varsity Online, The University of Toronto, 120(34), February, 1999
14. Callois, R. *Man, Play and Games.* The Free Press, 1961

15. Cassell, J., et al. Embodiment in conversational interfaces: Rea, *Proceedings of CHI*, 520-527, 1999

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

17. Cassell, J., et al. (eds), *Embodied Conversational Agents*, MIT Press, 2000

18. Cassell, J., et al. BEAT: the Behavior Expression Animation Toolkit, *Proceedings of SIGGRAPH*, 477-486, 2001

19. Cohen, P.R, et al. Quickset: Multimodal interaction for distributed applications, *Proceedings of the International Multimedia Conference,* ACM Press, 31-40, 1997

20. Corradini, A. and Cohen, P.R. On the Relationships among Speech, Gestures, and Object Manipulation in Virtual Environments: Initial Evidence, *Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, 52-61, 2002

21. Corradini, A., et al. Towards Believable Behavior Generation for Embodied Conversational Agents, *Proceedings of the International Conference on Computational Science (ICCS),* Lecture Notes in Artificial Intelligence 3038, Springer Verlag, 913-918, 2004

22. Fiske, S.T., et al. Social Cognition. McGraw Hill, 1991

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

24. Johnson, W.L., et al. Pedagogical Agents on the Web, *Proceedings of the International Conference on Autonomous Agents*, 283-290, 1999

25. Johnson, W.L., et al. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments, *International Journal of Artificial Intelligence in Education* 11, 47-78, 2000

26. Johnston, O., and Thomas, F. *The Illusion of Life*, Walt Disney Production, 1981

27. Katz, B. Annotating the World Wide Web using natural language, *Proceedings of the 12th International Conference on Information and Knowledge Management*, 2003

28. Loyall, A.B. Believable Agents: Building Interactive Personalities. PhD thesis, Technical Report CMU-CS-97-126, Carnegie Mellon University, 1997

29. Marriott, A., et al. VHML - Directing a Talking Head, *Proceedings of the International Conference on Active Media Technology*, 90-100, 2001

30. Massaro, D.W., et al. Development and Evaluation of a Computer-Animated Tutor for Language and Vocabulary Learning*, Proceedings of the 15th International Congress of Phonetic Sciences*, 2003

31. Nass, C., et al. Truth is beauty: Researching embodied conversational agents, In: Cassell, J. et al. (eds.), *Embodied Conversational Agents*, 374-402, 2000

32. Okonkwo, C., and Vassileva, J. Affective Pedagogical Agents and User Persuasion, In: Stephanidis, C. (eds.), *Proceedings of Universal Access in Human-Computer Interaction*, 2001

33. Oviatt S.L. Multimodal interfaces. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, In: Jacko, J. et al. (eds.), 286-304, 2003

34. Pazzani, M.and Billsus, D. Adaptive Web Site Agents*, Proceedings of the 3rd International Conference on Autonomous Agents*, 1999

35. Pelachaud, C., et al. Embodied Contextual Agent in Information Delivering Application, *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 2002

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

37. Picard, R. *Affective Computing*, MIT Press, 1997

38. Prensky, M. *Digital Game-Based Learning*. McGraw Hill, 2001

39. Reeves B., and Nass, C., *The Media Equation: how people treat computers, televisions and new media like real people and places*, Cambridge Univ. Press, 1996

40. Waibel, A, et al. Multimodal Interfaces, *Artificial Intelligence Review,* 10(3-4):299-319, 1996

41. Zheng, Z. AnswerBus Question Answering System, *Proceedings of the Human Language Technology Conference,* 2002