Providing Computer Game Characters with Conversational Abilities

Joakim Gustafson[1], Johan Boye[1],
Morgan Fredriksson[2], Lasse Johanneson[2], and Jürgen Königsmann[2]

[1] Voice Technologies, TeliaSonera, Rudsjöterrassen 5, 136 80 Haninge, Sweden
{joakim.gustafson, johan.boye}@teliasonera.com
[2] Liquid Media, Skånegatan 101, 116 32, Stockholm, Sweden
{morgan, lasse, jurgen}@liquid.se

Abstract. This paper presents the NICE fairy-tale game system, which enables adults and children to engage in conversation with animated characters in a 3D world. In this paper we argue that spoken dialogue technology have the potential to greatly enrichen the user's experience in future computer games. We also present some requirements that have to be fulfilled to successfully integrate spoken dialogue technology with a computer game application. Finally, we briefly describe an implemented system that has provided computer game characters with some conversational abilities that kids have interacted with in studies.

1 Introduction

The text adventure games of the 70's could achieve a limited sense of omniscience, since their goal-oriented users wanted to be immersed into the adventure, which refrained them from trying to deceive the system. The immersion was limited, due to the systems' limited understanding capabilities. Paradoxically, today's commercial 3D adventure games have even more limited input understanding capabilities – only allowing its users to navigate in the 3D world, selecting objects via mouse input and selecting what their avatar should do or say next from predefined menus. These computer games provide an excellent application area for research in spoken dialogue technology. Speech input is already used in some commercial computer games (e.g. Lifeline, 2004), but these do not support conversational interaction. More advanced spoken dialogue have the potential to greatly enrichen computer games. For example, it would allow players to refer to past events and to objects currently not visible on the screen, as well as interacting socially and negotiating solutions with the game characters.

The NICE project aims at to providing users with an immersive dialogue experience in a 3D fairy-tale game, engaging in multi-party dialogue with animated conversational characters. Spoken and multimodal dialogue is the user's primary vehicle of progressing through the story, and it is by verbal and non-verbal communication that the user can gain access to the goals and desires of the fairy-tale characters. This is critical as the characters ask the users to help them in solving problems. These problems either relate to objects that have to be manipulated or information that has to be retrieved from other fairy-tale characters.

2 Background

Spoken dialogue systems have so far mostly been designed with an overall goal to carry out a specific task, e.g. accessing timetable information or ordering tickets [4, 35]. With task-oriented systems, it is possible to build domain models that can be used to predefine the language models and dialogue rules. The existence of predefined tasks makes it rather straightforward to evaluate the performance of the dialogue system. Some dialogue systems have aimed to present its users with an engaging and entertaining experience, without the presence of an external predetermined task. Conversational kiosks, such as August [22] and MACK [12], encourage users to engage in social dialogues with embodied characters. Some spoken dialogue systems have addresses the problem of managing conversational speech with animated characters that reside in a 3D-world, e.g. the Mission Rehearsal Exercise system from the USC Institute of Creative Technologies, a system that also allow for multi-party dialogue [32].

Over the recent years interactive story-telling research systems have been developed that in some cases allow linguistic input. Hayes-Roth [24] lists a number of principles that are important for interactive story-telling systems. The user has to be given an illusion of immersion by participating in an interesting story where they feel that they are actively participating by interacting with the characters in a meaningful and natural way. Young [34] suggests that the drama manager of the system should put a limit to the user's actions by not allowing interference that violates the overall narrative plan. Most interactive games developed so far allow users to intervene in the storytelling by acting on physical objects on the screen using direct manipulation [10, 34]. Moreover, some systems allow users to interact with characters by means of written text input [28], while others explored using a speech interface [10].

3 Conversational Skills

Humans who engage in face-to-face dialogues use non-verbal communication such as body gestures, gaze, facial expressions and lip movements to transmit information, attitudes and emotions. If computers are to engage in spoken dialogue with humans it would seem natural to give them the possibility to use both verbal and non-verbal communication. Verbally, they have to be able to communicate their goals and plans to the user, and they should be able to cooperate with the user to solve problems. In order to convey personality and to build a collaborative trusting relationship with the users, the characters also have to be able to engage in socializing small talk. In order to be able to coordinate their action towards a goal that is shared with the user, the characters have to be able to collaborate with the user [15, 21]. The characters also have to be able to engage in grounding dialogue with the users to be able to certify that they have understood what the user wants them to do. In conversation the coordination of turns is crucial, and it is regulated by a number of turn management subfunctions that can be expressed verbally or non-verbally [3]. There are two simultaneous information channels in a dialogue: the information channel from the speaker, and the back-channel feedback from the listener. The back-channel feedback indicates attention, feelings and understanding, and its purpose is to support the interaction. It has been argued that dialogue systems should be able to provide positive feedback in successful contexts and negative feedback when problems have been detected [8]. Initial cue words can be used to facilitate grounding by providing information on the speaker's orientation towards the content of the previous turn [9]. Disfluencies like filled pauses may be indicators of problems in dialogue, but initial fillers are used to manage turn taking and both filled and silent pauses are used to indicate feeling-of-knowing [7].

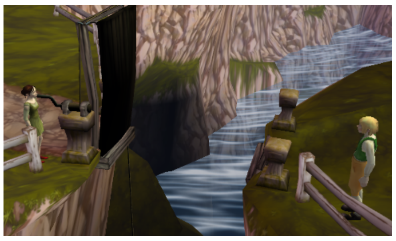
Animating the face brings the embodied character to life, making it more believable as a dialogue partner. Facial actions can be clustered according to their communicative functions in three different channels: the phonemic, the intonational and the emotional [18]. *The phonemic channel* is used to communicate redundant and complementary information in what is being said. *The intonational channel* is used to facilitate a smooth interaction. Facial expressions, eyebrow raising and head nods can be used to communicate the information structure of an utterance [11, 30]. *The emotional channel* is used to increase the animated character's social believability. There are *display rules* that regulate when speakers show emotions. These rules depend on the meaning the speaker wants to convey, the mood of the speaker, the relationship between speaker and listener and the dialogue situation [17]. Gaze indicates three types of mental processes: spontaneous looking, task-relevant looking and looking as a function of orientation of thought [25]. Thus, in conversation gaze carries information about what the interlocutors are focusing on, the degree of attention and interest during a conversation, to regulate the turn-taking, to refer to visible objects and to display emotions or to define power and status. Pelachaud et al. [31] described a facial animation system that among other things could display different gaze patterns, and the BEAT system uses gaze, head nods and eyebrow-raising for turn-handling [11]. Finally, turn-handling gaze can be used to indicate who is talking in multi-party dialogues such as virtual conferencing [16].

4 The NICE Fairy-Tale Game Scenario

The fairy-tale domain was chosen because of its classic themes and stereotypical characters, are well known to most adults as well as children. So far two scenes have been implemented, see Fig. 1. There are two main characters in the system: the helper Cloddy Hans, who has been introduced to facilitate progression through the story and gatekeeper Karen, who is introduced as an obstacle in the story. Personality traits are not explicitly modeled, but they are rather used as guidance in the design of the characters to ensure that their behaviors are perceived by the users as compatible with their intended personalities. Personality is conveyed by modes of appearance, actions, wording, speaking styles, voice quality, and non-verbal behavior. In order to match the characters' different roles in the game, the output behavior of the two characters have been designed to display these quite different OCEAN personality traits[29]: Cloddy Hans is *Dunce, Uncertain, Friendly, Polite, Calm* and *Even tempered*, while Karin *is Intellectual, Frivolous, Self-confident, Unfriendly, Touchy* and *Anxious.*

Cloddy Hans and the fairy-tale machine

The user meets Cloddy Hans in H. C. Andersen's study, where there is fairy-tale machine and a shelf with fairy-tale objects. The objects have to be put in one of several icon-labeled slots in the machine in order to construct a new story and thereby get transferred into the fairy-tale world. This introduction scene thus develops into a straightforward "put-that-there" game, where the system is able to anticipate what the user will have to say to solve it. The real purpose of the first scene is not to solve the task, but to engage in a collaborative conversation where the player familiarises himself with the possibilities and limitations of the spoken input capabilities.



Karen and Cloddy Hans at the drawbridge

The fairy-tale world is a large 3D virtual world, where the user and Cloddy Hans land on a small island, where they are trapped. A deep gape separates them from the rest of the world. There is a drawbridge in the gap, operated by Karin, who has the gatekeeper role in the scene. She will only lower the drawbridge when offered something she finds acceptable in return, which she never does until the user's third attempt, thereby encouraging negotiative behavior. Furthermore, both Cloddy Hans and Karen openly show some amount of grudge against each other, with both characters occasionally prompting the user to choose sides.

Fig. 1. The first two scenes in the fairy-tale game

Narrative Progression

The two scenes described above contain certain key moments, *story-functional events*. The passing of such an event means that there has been a progression in the story (thus it is important that a story-functional event can not be undone). The first scene contains the following story-functional events: *Cloddy Hans introduces himself; Cloddy Hans introduces the plot; Cloddy Hans picks up an object for the first time; Cloddy Hans puts object number X in the fairy-tale machine; Cloddy Hans pulls the lever so that he and the user can enter the fairy-tale world.* Since it is impossible to retrieve an object from the machine, all put-object-in-machine events are story-functional. The second scene contains the following types of story-functional events: *Cloddy Hans introduces the fairy-tale world; Karin introduces herself; Cloddy Hans gives his opinion of Karin; Karin gives her opinion of Cloddy Hans; Karin informs the user that she demands payment in order to lower the drawbridge; Karin accepts an object and lowers the drawbridge; Cloddy Hans crosses the drawbridge and gives Karin the payment.* The knowledge about these story functional events are encoded into the scene descriptions that all character loads when a scene is initialized. This means that they can add goals on their agenda that leads to the realization of all story functional events in a scene. This also makes it possible for the helper character Cloddy Hans to guide the user when she gets stuck in a scene. Some of events involve more than one action or the exchange multiple pieces of information. In order for the introduction to be complete Cloddy Hans has to talk about his and the users name, age and health. There are also default objects and corresponding destination slots in the scene description that could be used by the system to suggest a possible next object to pick up and then where it could be placed.

5 The Output Capabilities of the Fairy-Tale Characters

The fairy-tale characters in the NICE system are able to generate both verbal and non-verbal behaviour. They have different roles in the game and consequently they have to be able to convey different personalities that match their respective roles. Charles and Cavazza [13] distinguish between two types of characters in their character-based story telling system - *feature characters* and *supporting characters*. In the Nice fairy-tale game a third kind of character have been added - a *helper character*. This means that there are these three types of characters in the fairy-tale world, that require different levels of conversational abilities:

Helper Character - A character that guides and helps the user throughout the whole fairy-tale game. *Cloddy Hans* is a friendly character with no long-term goals for himself, other than doing what the user asks him to. Helper characters need conversational capabilities allowing both for grounding and cooperation, and for dialogue regulation and error handling. They need to have knowledge of all plots and subtasks in the game. Finally they need simple visual perception so that they can suggest actions that involves objects in the scene that the user have not noticed yet, and they have to be aware of the other characters actions as well as their verbal output.

Feature Characters - Characters that has a key function in the plots. *Karen* is a feature character that has a *Gatekeeper* function in the second scene. She is a selfish character with goals of her own. She will not help the user unless she gets a reward. Feature characters need less cooperative and grounding conversational abilities, since they have goals of their own that they simply want to convey to the user. However, they need dialogue regulation and error handling capabilities. They only need knowledge about the plots and subtasks they appear in, and they have to be aware of the other characters actions as well as their verbal output.

Supporting Characters - Characters that only tell the pieces of information needed for the plot, but that are not willing to engage in conversation with the user. Supporting characters only need to be provided with the verbal capabilities needed to convey the information they are supposed to communicate to the user. Apart from these they only need to be provided with verbal utterances like “I don’t want to chat with you”. They only need knowledge about the subtasks they are supposed to comment on, and they may be aware of the other characters actions as well as their verbal output. *Thumbelina* is added as a non-verbal supporting character that uses the default objects and destinations in the first scene in order to be able to point at the slot where she thinks a certain object should be placed. If the user gets Cloddy Hans to put it in another slot she shows her discontent with large emotional body gestures.

The fairy-tale characters are able to talk about the plots and scenes, as well as their own plans and to goals that relate to these. When characters first meet the user they are able to engage in formalized socializing small talk. In later phases they are still able to respond to social initiatives from the users, but without goals of their own to pursue the social topic. The characters are also provided with general dialogue regulating speech acts that they can use in all scenes: *Plan Regulating* (e.g. agree, ask for request), *Error Handling* (e.g. report not hearing, asking for clarification), *Turn Han-*

dling (e.g. floor holders), *Attitudal Feedback* (positive or negative feedback), *Discourse Markers* (respond to unexpected info), and *Extralinguistic sounds* (clear throat, exhalation, laughter, sigh). In order to be able to talk about the plot, their goals and plans, the fairy-tale characters have also been provided with a number of task oriented plot dependent speech acts: *Introduction and explanation of the plot, Initiatives that serve to fulfill the characters' plan or long-term goals, Requests for instructions, Responses to instructions from the user*, *Stating intentions, plans and goals***.**

The main characters are able to perform the actions needed to progress through the plots of the game. In order to be reactive they are also able to generate gestures as a result of user input and events in their environment. These reactions are either displays of attitude, state of mind, turn regulation gestures or attention gestures. The characters can also look at and point at interactive objects, non-interactive objects and landmarks in the 3D-world. They are able to walk between locations that are far apart. If the user has not engaged in interaction with the characters for a while they enter an idle state where they start off with small encouraging gestures, then after a while they indicate impatience by gazing around in the environment or displaying various idle gestures. All characters have been provided with a number of communicative gestures, as well as a number of simple, single body part animations that can be used to generate more complex multi body part gestures. This makes it possible to either play ready animations for communicative gestures or to generate animation lists consisting animation tracks on the individual body parts. Fig 2. below shows the different types of non-verbal behavior the characters are able to display.

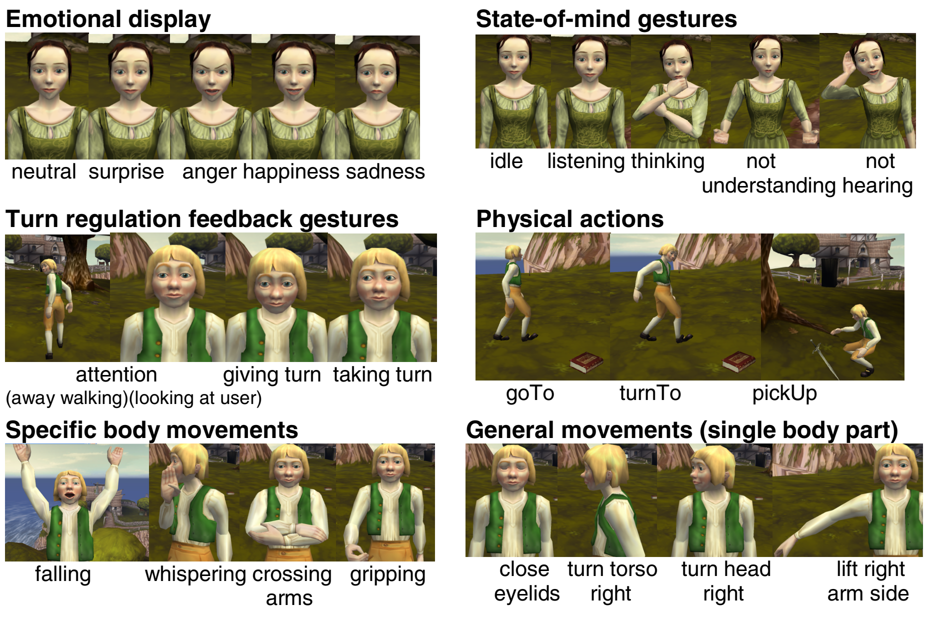


Fig. 2. The characters' different types of non-verbal behavior

The gestures, movements and actions of the different characters are used to convey their respective personalities. To make the characters' output behavior consistent, the body gestures, actions and idle behaviors of the two characters have been designed with their respective personality traits in mind. The manner in which characters move conveys their different personalities in the same way as their different speaking styles does. Chi et al [14] has developed a parametrisized system, EMOTE, that describes the manner of movements and Allbeck and Badler [2] describes an initial attempt to link the EMOTE parameters with the OCEAN personality parameters. If this linkage is applied to the two characters, they get the following EMOTE parameters with accompanying non-verbal behavior:

	Space	Weight	Time	Flow
	Direct: Single focus, e.g. he either looks bluntly at the user, or glances at the object that he or the user is referring to.	*Strong*: Powerful, having impact, e.g. he walks with determined steps.	*Sudden*: Hurried, e.g. he performs the actions asked for immediately	*Bound:* Controlled, restrained, e.g. he walks the shortest way to a location, and then he turns to the user, looking encouraging.
	Indirect: Multi-focus, e.g. doesn't look at the user for a very long time, before breaking their mutual gaze, letting her gaze wonder into the surroundings.	*Light*: Delicate, easily overcoming gravity, e.g. she walks about with light steps.	*Sustained*: Lingering, indulging in time, e.g. she tries to avoid to do what the users asks her	*Free*: Uncontrolled movement, e.g. she wanders about on her way to an location, looking as she doesn't quite know where she is heading

Fig. 3. The impact of the derived EMOTE parameters on the characters' non-verbal behaviors

To support the intended personalities of these characters, Cloddy Hans displays small, but slow and deliberate body gestures while Karen displays larger, and faster body gestures. The characters' different personalities are also conveyed by their different idle behaviors: Karin is not patient which is reflected by the fact that she enters her idle phase faster, and she lets her attention wander away from the user to the environment, and after a while she even walks away from the user. Cloddy is more calm and keeps his attention at the user. Finally, to give the characters basic simple perceptual abilities a number of reactive behaviors have also been added in the system:

Auditory Perception - is simulated by generating attention gestures that for example involve turning to the speaker. When user speech is detected the characters will turn to the active camera, and when there are multiple character speaking in a scene the other characters will turn towards the speaking character

Visual Perception - is simulated by generating attention gestures when the users starts gesturing or glancing at the object that the user has encircled. It is also simulated by adding triggers nearby interesting objects, and generating an appropriate attention gesture towards an object that the character walks by. It is also possible for the system to request a list of all objects that are visible (either on the screen or from a characters field of vision), and then request the character to turn to or talk about a found object.

Perception of Time - is simulated by letting a central server time-stamp all messages from both input and output modules, and by letting it generate timeouts that are used to manage the characters' idle behavior. The Animation system keeps track of all characters' current actions, in order to be able to change a certain character's behavior dependent on the current situation and to be able to coordinate different characters' simultaneous actions.

6 System Architecture

To make the animated fairytale characters appear lifelike, they have to be autonomous, i.e. they must do things even when the user is not interacting with them. At the same time they have to be reactive and show conversational abilities when the user is interacting with them. This means that the characters have to be able to generate carefully planned goal-oriented actions as well as very fast, less planned actions. In order to be able to build a system that can harness all these functionalities, an event driven, asynchronous, modular hub architecture was chosen, where a set of processes that communicate via message-passing over TCP/IP. Events from all servers are sent to a central hub, the Message Dispatcher server, (similar but simpler than OAA [27] or Communicator [1]). The central Message Dispatcher is responsible for coordinating input and output events in the system, by time-stamping all messages from the various modules. The behavior of the Message Dispatcher is controlled by a set of simple rules, specifying how to react when receiving a message of a certain type from one the modules. Since the Message Dispatcher is connected both to the input channels and the output modalities, it can increase the system's responsiveness by giving fast but simple feedback on input events. Timeouts from the Message Dispatcher are used to allow the system to have a perception of time, which is used to control the characters' idle behavior, and to let the dialogue managers take the imitative and generate suggestions of actions in cases where the users has not answered a request for the next action.

The spoken input is handled by a speech recognizer with statistical language models trained on 5600 user utterances from 57 users that interacted with a semi-automated version of the system (the wizard could correct the ASR-string if needed). A robust natural language understanding module has also been developed using this data[6]. To be able to provide the animated character with Swedish voices with natural voice quality and prosody, a unit selection synthesizer was developed in cooperation with KTH [23]. An important role of the synthesis component in the fairy-tale system is to convey the personality of the characters. To get to the different speaking styles, the voice talents were told to read the utterances in manners that matched the targeted personalities. This resulted in two voices with speaking styles that, among other things, differed in frequency range. They also differed in speaking rate and voice pitch. In order to accentuate these last two differences, all utterances were re-sampled changing speaking rate and voice pitch at the same time. All Cloddy's utterances were slowed down and all Karen's utterances were speeded up. This simple procedure had desired side-effects: apart from making Cloddy's voice slower it made him sound larger, and, apart from making Karen's voice faster, it made her sound younger. The personalities of the two characters were deliberately chosen so that this simple voice transformation would also make their voices more matching with the visual appearance of the two animated characters.

6.1 Dialogue Management

There are two dialogue managers in the NICE fairy-tale game system, one per fairy-tale character. The functionality of these two dialogue managers are somewhat different, reflecting the fairy-tale characters' different personalities. Moreover, the functionality of any dialogue manager varies over time, reflecting supposed changes in the characters' knowledge, attitudes and state of mind. However, when considered at an appropriate level of abstraction, most of the functions any dialogue manager needs to be able to carry out remain constant regardless of the character or the situation at hand. As a consequence, the dialogue management software in the NICE fairy-tale system consists of a *kernel* laying down the common functionality, and *scripting code* modifying the dialogue behavior as to be suitable for different characters and different situations[5]. This model of code organization is common in computer games[33].

The dialogue management kernel issues dialogue events at important points in the processing. Some kinds of dialogue events, the so-called external events, are triggered from an event in a module outside the dialogue manager (for instance, a recognition failure in the speech recognizer), whereas for others, the internal events, an internal event takes place within the dialogue kernel. There are e number of external dialogue event that the dialogue manager can receive: *BroadcastEvent* (some other character has said and done something), *GestureEvent* (the Gesture Interpreter has recognized a gesture), *ParserEvent* (the natural language parser has arrived at an analysis of the latest utterance), *PerformedEvent* (the animation system has completed an operation), *RecognitionFailureEvent* (the speech recognizer has detected that the user has said something, but failed to recognize it), *WorldEvent* (an event has occurred in the 3D world), and *TriggerEvent* (the animation system has detected that the character has moved into a trigger). There are also a number of internal dialogue events: *Al-readySatisfiedEvent* (a goal which already is satisfied has been added to the character's agenda), *CannotSolveEvent* (an unsolvable goal has been added to the character's agenda), *IntentionEvent* (the character has an intention to say or do something), *No-ReactionEvent* (the character has nothing on the agenda), *PossibleGoalConflictEvent* (a goal is added to the agenda, but the agenda contains a possibly conflicting goal), and *TimeOutEvent* (a timeout has expired). The kernel provides a number of operations through which the scripting code can influence the dialogue behaviour of the character. These are: *interpret an utterance* in its context; *convey* a dialogue act; *perform* an action; *add a goal* to the character's agenda; *remove a goal* from the character's agenda; *find the next goal* on the agenda, and *pursue a goal* on the agenda.

6.2 The Animation System

The *Animation System,* (see Fig. 4) is responsible for generating the character animations and actions. It is divided into two modules: The *Animation Handler* and the *Animation Renderer.*

6.2.1 The Animation Handler

The Animation Handler deconstructs action requests from the dialogue managers into sequences of more fine-grained animation instructions. For instance, a "go to the fairy-tale machine" request is translated into (1) change camera (2) walk to the machine (3) change camera again, and (4) turn to camera. These animation instructions are queued (there is one queue per fairy-tale character) and sent one at a time to the Animation Renderer. After successful execution of an instruction, the Renderer sends back a receipt, after which the next instruction is sent, and so on. Upon receipt of a speech-synthesis request from a dialogue manager, the Animation Handler instructs

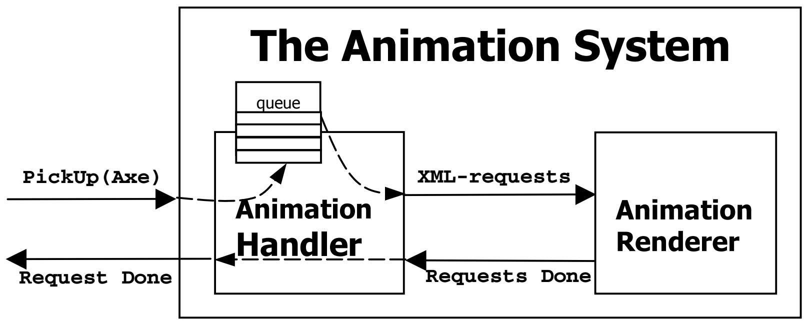


Fig. 4. The internal handling of requests to the Animation System and its place in the system

the speech synthesizer to generate a sound file with corresponding *lip-synchronization track*. The latter is a sequence of time-stamped animation instructions for the different facial movements needed to achieve lip-synchronization. If the character is walking or otherwise moving when the lip-synchronization instructions are rendered, the facial animations are blended with the bodily movements.

Within the Animation Handler, there is one synthesis queue per fairy-tale character. Each speech-synthesis request comes with a priority, needed to determine the correct action to take if the character is already talking (due to the event-based dialogue management method, this happens occasionally). If the priority of the incoming request is lower than that of the utterance currently being produced, the incoming request is ignored. If the priority of the incoming request is higher, the ongoing utterance is interrupted, and the new utterance is produced instead. If the two requests have equal priority, the incoming request is enqueued and produced after the ongoing utterance has finished. Synthesis requests with high priority typically concern replies to the user's utterance, requests with medium-high priority typically concern suggestions to the user on how to proceed (generated when the user has been silent for a while), and requests with low priority concern chit-chat.

6.1.2 The Animation Rendering System

The subsystems of the rendering system communicate with each other through façade classes with virtual interfaces [19]. The use of virtual facades makes it easy to switch between different implementations of a specific subsystem without affecting the other systems or applications using the game engine. The *Resource System* is a responsible for keeping track of all resources (like e.g. graphical meshes and animated models). All resources have been given a type and a name in order to make them unique and distinguishable from each other. The *Animation System* is responsible for creating and updating animated models. An *Animated Model* is a deformable graphical object built upon a hierarchy of frames. Each frame has a 3D-space transformation matrix representing its rotation and position relative the parent frame. An *Animation* is a named data set containing rotation, position, and scale values for a given frame and point in time. Animations can contain values for all of the frames in the hierarchy or just a single one. The graphical artist creating the animation decides which frames that are included in an animation. To be able to move different parts of the hierarchy simultaneously and independently, animated models can be ordered to play animations at separate *tracks*. Currently each model has 8 tracks, but this could easily be changed if there is a need for it. If animations played at different tracks affects the same set of frames, the resulting movement will be decided by the animation played at the track with the highest index.

It is the physics and collision systems that are responsible for real time simulation of the movement of walking characters, falling objects etc. The collision system is also responsible for handling some of the game logic controlling duties, such as triggering events when an object enters a specific area or picking out objects selected by the user mouse input. The animation system uses an externally developed collision and a physics system called Tokamak [26]. To speed up the complex calculations involved in realistically simulating collisions between objects, 3-dimensional mathematical shapes are used as simple collision primitives. Objects and characters are provided with one or more simple collision primitive. In addition to collision between these simple collision primitives, the system also supports collisions between primitives and arbitrary shaped geometries. For performance reasons only one complex collision object is allowed to be active at the time. This single complex collision geometry is normally used for the static game world environment. The complex collision geometry is automatically generated from the files that describe the visual appearance of the game world. The graphics designer has however the opportunity to exclude some parts of the visual geometry from the resulting collidable geometry. He can also add geometry that only will be collidable and not visible.

The XML interface to the rendering system includes the following actor commands: *ResetAnimController, ClearAnimationTrack, GetPostition, GoTo, turnTo, play*(single animation, a sound or Animation list with parameters for start percentage and speed*), PickUp, releaseHeldObject, Jump*. There are also a number of object commands: *GetPostition, SetPosition, Highlight, Render, PutInPlace, TogglePhysicalState*. The camera commands include: *setActiveCamera, InterpolateToCamera, SetTargetEntity*. Finally there are a number of other commands: *GetOnScreenObjects, SetInputReceiver, ExecuteCommandLine* (commands to the renderers built in Python-based script interpreter), *GetCurrentLevelName, SetCurrentLevel*.

7 User Evaluations

During 2004–2005, data was collected on several occasions using the NICE system at different stages during its development. The system could be run either in fully automatic mode or in supervised mode, in which a human operator had the possibility to intervene and modify the ASR result or select an appropriate output speech act or action for a character to perform next. This made it possible to develop the system in a data-driven, iterative fashion. 57 children (aged 8 to 15) interacted with the system, resulting in a human–computer dialogue corpus containing about 5,800 system turns and 5,600 user turns. The usability study showed that the addition of spoken and multimodal dialogue creates a positive user experience, and that it is possible to use spoken language as the main device for story progression. In the interviews, users unanimously reported that Cloddy Hans was a bit stupid, but kind, while Karen being rather the opposite. Personality differences were also found in analyses of the post-experiment questionnaires, where the user judged the how well different personality traits described the characters. Differences between Cloddy Hans and Karen were tested for significance using Wilcoxon Signed Ranks Test ($p<0.05$). It was found that users rated Cloddy Hans as more *Kind, Stupid, Lazy, Calm, Polite* and *Distressed*, while Karin was found to be more *Smart, Quick* and *Self-confident*.

The analysis of the users' interaction showed that there were significant differences in speaking rate were observed between the User-Cloddy dialogues and the User-Karin(-Cloddy) dialogues. In the first repetitive scene the users took more and more initiative and needed fewer and fewer turns for each object they put into the machine, but at the same time they talked slower and slower, to make sure that the sluggish Cloddy Hans would understand them. In the second scene, they started of speaking faster, but then they slowly began to talk slower again with Cloddy Hans for each turn during their exploration of the island. As soon as they started talking to Karin, they talked faster, but this time the actually increasing their speech rate for each turn. This could be because the interaction was faster, and because it felt more lively when Cloddy Hans came with side-comments during the negotiation between the user and Karin. It was probably also because Karin appeared to be smarter when she drove the dialogue, without showing any problems in understanding the user. Actually, she did not always have to understand what the user said since she could "see" that Cloddy had brought a certain object that she then simply could reject regardless of how the user presented it.

To conclude, the user study showed that it was possible to design characters, which were perceived as having fundamentally different personalities and conversational abilities, and that in three-party dialogue with several animated figures each character was regarded as a separate entity who did not always hear or understand the others. This made it possible to decrease the shortcomings of the speech recognizer by letting the system tell the users (via Karen) what they should say to Cloddy Hans in the next turn. That this seemingly simple trick "worked" is indicated by the fact that users rated Cloddy Hans as stupid and Karen as smart even though the trick was used in both directions. Finally, several users explicitly perceived shortcomings of the natural language interface as part of the game, constituting a new kind of interesting and engaging obstacle to overcome. They thought it could be an interesting mind puzzle to figure out how to talk and what to say in order to get Cloddy Hans to do what the wanted.

Acknowledgements

This research was carried out within the EU 5th framework project NICE (IST-2001-35293). The NICE homepage can be found at http://www.niceproject.com.

References

1. Aberdeen, J., Bayer, S., Caskey, S., Daminos, L., Goldschen, A., Hirschman, L., Loehr, D. and Trappe, H. (1999) Implementing practical dialogue systems with the DARPA Communicator architecture. Proc. IJCAI'99 workshop on knowledge and reasoning in practical dialogue systems,pp 81-88.

2. Allbeck, J. and Badler, N. “Representing and Parameterizing Agent Behaviours” In Life-Like Characters: Tools, Affective Functions, and Applications, Prendinger, H. and Ishizuka, M, Eds. Springer, Germany, 2004.
3. Allwood, J. “Reasons for management in dialog“, in Beun, R.J., Baker, M. and Reiner, M. (eds.) Dialogue and Instruction. Springer-Verlag.pp 241-50, 1995.
4. Aust, H., M. Oerder, F. Seide and V. Steinbiss (1995). The Philips automatic train timetable information system. Speech Communication 17(3-4): 249-262.
5. Boye, J., and Gustafson, J. (forthcoming) “How to do dialogue in a fairy-tale world”, a demo session paper at the sixth SIGdial Workshop on Discourse and Dialogue, Lisabon, 2005.
6. Boye, J, Gustafson, J. & Wirén, M. (Forthcoming) “Robust spoken language understanding in a computer game”,J. of Speech Communication,forthcoming special issue on spoken language understanding.
7. Brennan, S. “Processes that shape conversation and their implications for computational linguistics,” Proceedings, 38th Annual Meeting of the ACL. Hong Kong, 2000.
8. Brennan, S. and Hulteen, E. “Interaction and feedback in a spoken language system: a theoretical framework,” Knowledge-Based Systems(8): 143-151, 1995.
9. Byron, D. and Heeman, P. “Discourse Marker Use in Spoken Dialog”, In Proceedings of the 5th Eurospean Conference On Speech Communication and Technology, Rhodes, Greece, September 1997.
10. Cavazza, M., Charles, F. and Mead S. J. (2002). Character-based interactive storytelling. IEEE Intelligent Systems, Special issue on AI in Interactive Entertainment,, pp. 17-24.
11. Cassell, J., Vilhjálmsson, H. & Bickmore, T. “BEAT: The Behavior Expression Animation Toolkit”. Proceedings of SIGGRAPH ’01, Los Angeles, CA, 2001, pp. 477–486, 2001
12. Cassell, J., T. Stocky, T. Bickmore, Y. Gao, Y. Nakano, K. Ryokai, D. Tversky, C. Vaucelle and H. Vilhjlmsson (2002). MACK: Media lab Autonomous Conversational Kiosk. Imagina 02. Monte Carlo.
13. Charles, F. and Cavazza, M. (2004) Exploring the scalability of character-based storytelling. Proc. ACM Joint conference on autonomous agents and multi-agent systems, New York, USA.
14. Chi, D., Costa, M., Zhao, L. and Badler, N. “The EMOTE Model for Effort and Shape,” Proceedings of SIGGRAPH 2000, ACM Computer Graphics Annual Conference, New Orleans, Louisiana, 23-28 July, 2000, pp. 173-182.
15. Clark, H. “Managing problems in speaking”, Speech Communication, 15:243-250, 1994.
16. Colburn, A., Cohen, M. & Drucker, S. (2000) “The Role of Eye Gaze in Avatar Mediated Conversational Interfaces”. MSR-TR-2000-81. Microsoft Research, 2000.
17. Ekman, P. *Emotion in the human face*. New York: Cambridge University Press, 1982.
18. Ekman, P. “About brows: Emotional and conversational signals”. In: Cranach, M. von, Foppa, K., Lepenies, W. & Ploog, D. (eds.), Human Ethology: Claims and Limits of a New Discipline: Contributions to the Colloquium, Cambridge, Cambridge University Press, pp. 169–202, 1979
19. Gamma E., Helm R., Design patterns: elements of reusable object-oriented software, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1995.
20. Goodwin, C. “Conversational Organization: interaction between speakers and hearers,” New York/London, Academic Press, 1981.
21. Grosz, B. and Sidner, C. “Attention, Intention, and the Structure of Discourse,” Computational Linguistics 12(3), pp. 175–204, 1986.
22. Gustafson, J. and L. Bell (2000). Speech technology on trial - Experiences from the August system. Natural Language Engineering 6(3-4): 273-286.

23. Gustafson, J. and Sjölander, K., (2004). "Voice creation for conversational fairy-tale characters." In Proceedings of the 5th ISCA Speech Synthesis Workshop. Pittsburgh.
24. Hayes-Roth, B. "Character-based Interactive Story Systems," IEEE Intelligent Systems and Their Applications 13.6: pp 12-15, 1998.
25. Kahneman, D. *Attention and Effort*. Prentice–Hall, Englewood–Cliffs, New Jersey, 1973.
26. Lam D, "Tokamak Game Physics SDK", http://www.tokamakphysics.com/, 2004.
27. Martin, D. and Cheyer, A. and Moran, D. (1999) The Open Agent Architecture: a framework for building distributed software systems. Applied Artificial Intelligence, vol. 13, no. 1-2, pp. 91-128, January-March 1999.
28. Mateas, M. and A. Stern (2002). Architecture, authorial idioms and early observations of the interactive drama Facade. Technical report CM-CS-02-198.
29. McCrae, R. and Costa, P. "Toward a new generation of personality theories: Theoretical contexts for the five-factor model," In J. S. Wiggins (Ed.), The five-factor model of personality: Theoretical perspectives (pp. 51-87). New York: Guilford, 1996.
30. Pelachaud, C. & Prevost, S. "Sight and Sound: Generating Facial Expressions and Spoken Intonation from Context". Proceedings of the second ESCA Workshop on Speech Synthesis, New Paltz, NY, USA, September 1994, pp. 216–219, 1994.
31. Pelachaud, C., Badler, N. & Steedman, M. "Generating Facial Expressions for Speech". Cognitive Science 20(1), pp. 1–46, 1996.
32. Traum, D. (2004) Issues in multi-party dialogues. In Dignum (ed), Advances in agent communication, Lecture notes in artificial intelligence 2922, pp. 201-211, Springer-Verlag.
33. Varanese A. and LaMothe, A. (2003) Game scripting mastery. Premier Press.
34. Young, R. M. (2001). An overview of the Mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. Working notes of the AAAI spring symposium on Artificial intelligence and interactive entertainment.
35. Zue, V., J. Glass, D. Goodline, H. Leung, M. Phillips, J. Polifroni and S. Seneff (1991). Integration of speech recognition and natural language processing in the MIT voyager system. Proc. ICASSP'91. Toronto.

An example dialogue in the second scene with actual dialogue excerpts from three children's interactions	
user	where are we?
cloddy	now we are in the fairy-tale world…what should we do?
user	please walk over and pick up the sword
cloddy	good idea *[cloddy walks over to the sword]*
user	walk over to the girl over there
cloddy	her name is Karin *[cloddy walks over to the bridge where Karen stands on the other side]*
user	could you please talk with Karin
cloddy	why don't you talk with her instead … she usually is a bit ill-tempered
user	okey *[the camera moves from Cloddy to Karin]*
karin	who are you?
user	Johanna .. and you
karin	my name is Karin…how are you?
user	fine how are you
karin	well so so ... I'm an orphean and I hardly have any friends
user	would you like to become my friend?
karin	yes!
user	could you please lower the drawbridge for us?
karin	if I am to lower the drawbridge i want something in return
cloddy	she is easily bribed *[from the side, without camera turning away from Karin]*
user	how about this sword?

karin	you can keep that old junk for yourself, that's nothing for me
cloddy	she's so grumpy *[from the side, without camera turning away from Karin]*
user	but what do you want then?
karin	I thought I saw something shining falling down at the same time as the schmuck
user	you can have a ruby .. I'll return with something nice for you
karin	why do you keep dragging along that Cloddy Hans figure, by the way
user	he refuse to leave
cloddy	mmm (with negative prosody) *[the camera turns to Cloddy]*
user	drop the sword *[cloddy drops the sword]*
user	could you please pick up the emerald
cloddy	ok I'll walk over to the jewels *[walks over to the emerald and picks it up]*
user	go back to Karin *[cloddy walks to Karin where the camera swings over to Karin]*
karin	what do you have for me?
user	an emerald
karin	now you're talking
user	could you lower the bridge now?
karin	okey *[Karin lowers the drawbridge]*
cloddy	Hurray
user	walk over the bridge *[Cloddy walks over the bridge]*
user	now give her the emerald, after all she did lower the bridge
cloddy	should we really give it to her - shouldn't we keep it?
user	give it to her even though she was rude to you
cloddy	but she is only a little runt
user	yeah..but she DID lower the bridge
karin	well if only you had asked nicely I would have let you over anyway
user	but you said that you needed the something nice for you to be able to lower the bridge!?!
karin	I don't care!
user	if you want the emerald then you'll have to apologize!