NICE project (IST-2001-35293)

Natural Interactive Communication for Edutainment

NICE Deliverable D3.3

Analysis and specification of cooperation between input modalities and cooperation between output modalities

March 24, 2003

Authors

Jean-Claude Martin[4], Niels Ole Bernsen[1], Reinhard Blasig[2], Johan Boye[3], Stéphanie Buisine[4], Laila Dybkjær[1], Morgan Fredriksson[5], Michel Généreux[1], Joakim Gustafson[3], Ulrik Lindahl[5], Manish Mehta[1]

1: NISLab, Odense, Denmark, 2: Scansoft Aachen GmbH, Germany, 3: Telia Research AB, Farsta, Sweden,

4: LIMSI-CNRS, Orsay, France, 5: Liquid Media, Stockholm, Sweden

Project ref. no.	IST-2001-35293
Project acronym	NICE
Deliverable status	Internal
Contractual date of delivery	Month 12 (28 Feb 2003)
Actual date of delivery	March 15, 2003
Deliverable number	D3.3
Deliverable title	Analysis and specification of cooperation between input modalities and cooperation between output modalities
Nature	Report
Status & version	Final
Number of pages	43
WP contributing to the deliverable	WP3
WP / Task responsible	LIMSI-CNRS
Editor	Jean-Claude Martin
Author(s)	Jean-Claude Martin, Niels Ole Bernsen, Reinhard Blasig, Johan Boye, Stéphanie Buisine, Laila Dybkjær, Morgan Fredriksson, Michel Généreux, Joakim Gustafson, Ulrik Lindahl, Manish Mehta
EC Project Officer	Mats Ljungqvist
Keywords	Architecture, multimodal input, multimodal output.
Abstract (for dissemination)	This deliverable is a progress report that aims at providing descriptions and informal specification of cooperations between input modalities (multimodal behavior that we might expect from the users) and cooperations between output modalities (multimodal behavior that could be included in the virtual characters). It also describes the architecture that we propose to develop for the NICE system in order to enable such input and output multimodality.

Table of Contents

1 Goal of this deliverable 4
2 Architecture 4
2.1 Global architecture 4
2.2 Message dispatcher 7
2.3 Recognition and presentation layer 8
2.4 Description of each module of the simulation layer 11
2.5 Description of each module of the character module layer 14
3 Input 18
3.1 NISLab data collection on verbal conversations with a 2D talking head of Andersen 18
3.2 LIMSI-CNRS data collection on multimodal input behavior when interacting with 2D cartoon like characters 20
3.3 Telia data collection of children's speech 22
3.4 Telia data collection using a 2D virtual Cloddy Hans 22
4 Informal specification of multimodal output 24
4.1 Suggestions grounded on observations from video of HCA behavior 25
4.2 Suggestions grounded on the LEA cartoon-like agents 31
4.3 Mark-up language 37
5 Future directions 42
6 References 43

Analysis and specification of cooperation between input modalities and cooperation between output modalities

1 Goal of this deliverable

This deliverable is a progress report that aims at providing descriptions and informal specification of cooperations between input modalities (multimodal behavior that we might expect from the users) and cooperations between output modalities (multimodal behavior that could be included in the virtual characters). We start by describing the architecture that we propose to develop for the NICE system in order to enable such input and output multimodality. Section 2 describes the global architecture of the system. Section 3 provides suggestions regarding the input side (preliminary ideas and issues on the multimodal behavior that may be produced by the users of the future NICE system). Section 4 summarizes brainstorming ideas and suggestions that have been collected regarding the output side (potentially interesting behaviors for the embodied agents and first proposals regarding the format of messages to be used to drive the multimodal output behavior of the agents).

The goal of this report is to drive the specification and development of preliminary software development for experimental studies which will in turn provide more concrete data for the requirements specification regarding the cooperation between input modalities and between output modalities.

2 Architecture

2.1 Global architecture

A high-level sketch of the NICE system architecture is presented in deliverable D6.1 *System architecture and module communication* in autumn 2002. Since then, the three layers of the NICE architecture, i.e. the recognition and presentation layer, the simulation layer, and the character module layer, respectively, have been refined at a second level of detail so that we now have a common understanding in the project about the main modules that will be implemented and the information that will be exchanged between the main modules. Figure 2.1.1 shows a slightly revised version of the architecture presented in D6.1. The figure shows the three architecture layers including a high-level system component breakdown. Figure 2.1.2 provides a more detailed presentation of the main system components and the main information flow among them.

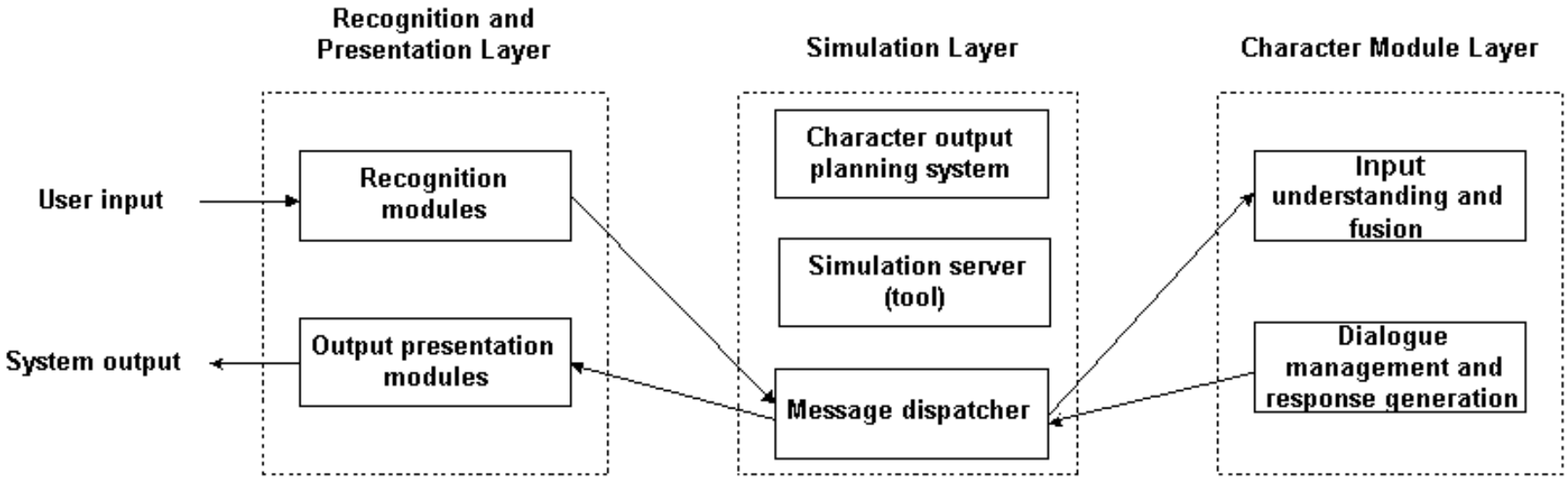

Figure 2.1.1. High-level sketch of the NICE architecture.

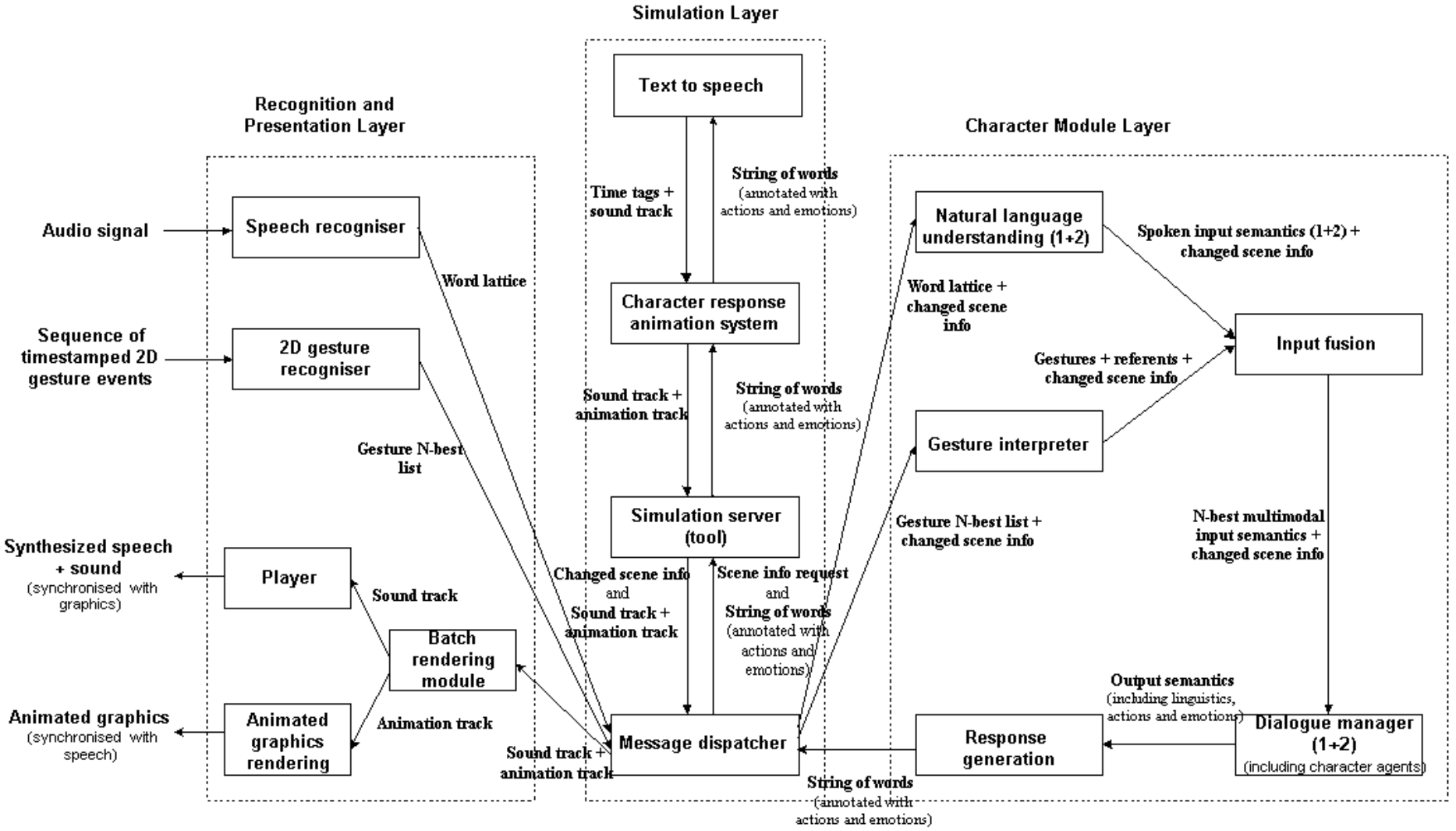

Figure 1.1.2. NICE architecture showing main components and main information exchanges. (1+2) means that there will be an HCA version of the component and a fairy tale version of the component.

In addition to the system input/output shown in Figure 2.1.2, the NICE system may take input from other devices as well, such as registration of the fact that a user is now sitting in the chair in front of the system, eliciting open microphone and/or speech recognition initialisation. Also, Figure 2.1.2 does not show that expectations will probably be passed between some of the modules, such as from the Dialogue manager to the Speech recogniser and the Natural language understanding module(s). Furthermore, there may be additional information passed around in the system, such as feedback on whether a requested action could be successfully performed. The precise decisions on these issues will be made on an experimental basis during further analysis, specification, and development. Time information will be attached to each piece of information exchanged between modules.

As already explained in D6.1, we want to leave open the possibility of running system components distributed on several computers. We also want to leave open the possibility of process communication over the internet as already used in certain types of game. Therefore, we have chosen TCP/IP as the standard communication protocol. For distributed processing we could also, in principle, have chosen CORBA or DCOM. However, from previous projects the consortium has rather negative experience with CORBA which requires rare expert knowledge to be successful and does not support integration of modules written in certain programming languages, such as Prolog, and neither CORBA nor DCOM are competitors to TCP/IP socket connections as regards internet communication.

On top of the TCP/IP connection, we use XML for wrapping the information to be exchanged. The choice of TCP/IP combined with XML should facilitate integration of each partner's modules. TCP/IP is widespread and well-known also by the project partners. XML is becoming a standard and is used in several other human-communication systems comparable to NICE [Dagstuhl 2001, AAMAS 2002, PRICAI 2002]. The combination of TCP/IP with XML would seem to be a strong and promising choice for the NICE system.

The following sections describe each module in Figure 2.1.2 in terms of:

- messages received as input to the module,
- overall description of the processing done in the module,
- output messages sent from the module,
- an example of output produced by the module.

2.2 Message dispatcher

- *Function* : this module is in the Simulation layer and is central for the system communication by routing messages between the modules. The dispatcher adds timestamps to every message and uses the simulation system to decide which messages to send to which agents. For instance, an agent will be informed when the state of some object changes only if it is sufficiently close. The message dispatcher has similarities to what is called a *facilitator* in other multi-agent systems.
- *Input* : a message sent by a module (i.e. connection-related message, message to be sent to another module).
- *Internal data and models used by this module* : routing information (e.g. directories).
- *Output* : forwarded message to the specified module.

2.3 Recognition and presentation layer

2.3.1 Speech recogniser

The speech recognition engine takes acoustic input in the form of alaw- or µlaw-coded sample data. The sampling rate used within the NICE project is 16kHz. For the decoding process, the recognizer also needs the following resources: an acoustic model (as provided by ScanSoft), lexicons and language models. All of these resources may be exchanged in between the recognition processes for separate utterances. In particular, it is possible to modify the lexicon and language model depending on the current dialog state, which can be done to optimize the modeling of an expected utterance.

The output of the recognizer will be

1. N-best paths providing recognition results on word level. This data is coded in the form of C structures and is returned into linear buffers which have to be allocated by the application calling the SpeechPearl recognizer. For each recognized word, the result indicates start and end time (relative to utterance start), word spelling and confidence value. More details about the result structures can be found in the SpeechPearl reference manual.
2. Word graphs (lattices) that can be logged into files by switching the SpeechPearl parameter `logWordGraph` on. The word graph files mainly contain one tabular structure per utterance which represents a single word hypothesis per line. The columns of the table indicate for the respective word
    a. the start node within the word graph
    b. the end node within the word graph
    c. the spelling
    d. the acoustic score
    e. the start time
    f. the end time
    g. the confidence measure

Example:

```
BEGIN_LATTICE
1     2     #PAUSE#          92.90        0    270   -1.0000
2     3     hello            125.63     270    640   1.0000
3     4     world            74.82      640    860   1.0000
4     5     #PAUSE#          383.96     860   1980   -1.0000
END_LATTICE
```

The SpeechDetector is a separate module that signals start of speech to the speech recognizer and to the message dispatcher. Together with the relative timing information in the Nbest result or the word graph, the message dispatcher can thus determine the exact absolute start and end times of recognized words. This allows input fusion to relate the timing of speech input and gesture input.

2.3.2 2D gesture recogniser

- *Function* : Low-level geometric processing of gestural data.
- *Input* : sequences of (timestamps, x, y) produced by a 2D input device (e.g. Tactile screen)
- *Internal data and models used by this module* : gesture models as observed in Wizard of Oz studies, heuristic rules.
- *Output* : n-best list of gesture shape, direction and size
- *Exemple of message produced by this module* :

```
<recognisedGesture>
  <hyp n="1">
    <score>0.8</score>
    <begin>...</begin>
    <end>...</end>
    <shape>line<\shape>
    <2DboundingBox>xmin, ymin, xmax, ymax<\2DboundingBox>
    <direction>…<\direction>
  <\hyp>
  <hyp n="2">
    <score>0.6</score>
    <begin>...</begin>
    <end>...</end>
    <shape>arrow<\shape>
    <2DboundingBox>xmin, ymin, xmax, ymax<\2DboundingBox>
    <direction>…<\direction>
  <\hyp>
<\recognisedGesture>
```

2.3.3 Batch rendering

- *Function* : decides what is visible and audible for the user ; prepares 3D and audio data based on the user perspective and passes it along to the audioplayer and 3D rendering system
- *Input* : Sound and animation tracks along with changes in the world state and user perspective
- *Output* : passes what is visible and audible for the user along to the audio player and graphical renderer : keyframed animations in the .x format and multi-channeled audio in the .wav format

2.3.4 Audioplayer

- Mixed multi channeled 3D positioned audio .
- *Function* : audio player
- *Input* : .wav
- *Output* : Mixed multi channeled 3D positioned audio

2.3.5 Animated graphics rendering

- *Function* : 3D Renderer
- *Internal data and models used by this module* : .x and .tiff or .png
- *Output* : A graphical representation of the world

- *Exemple of graphical representation produced by this module* :

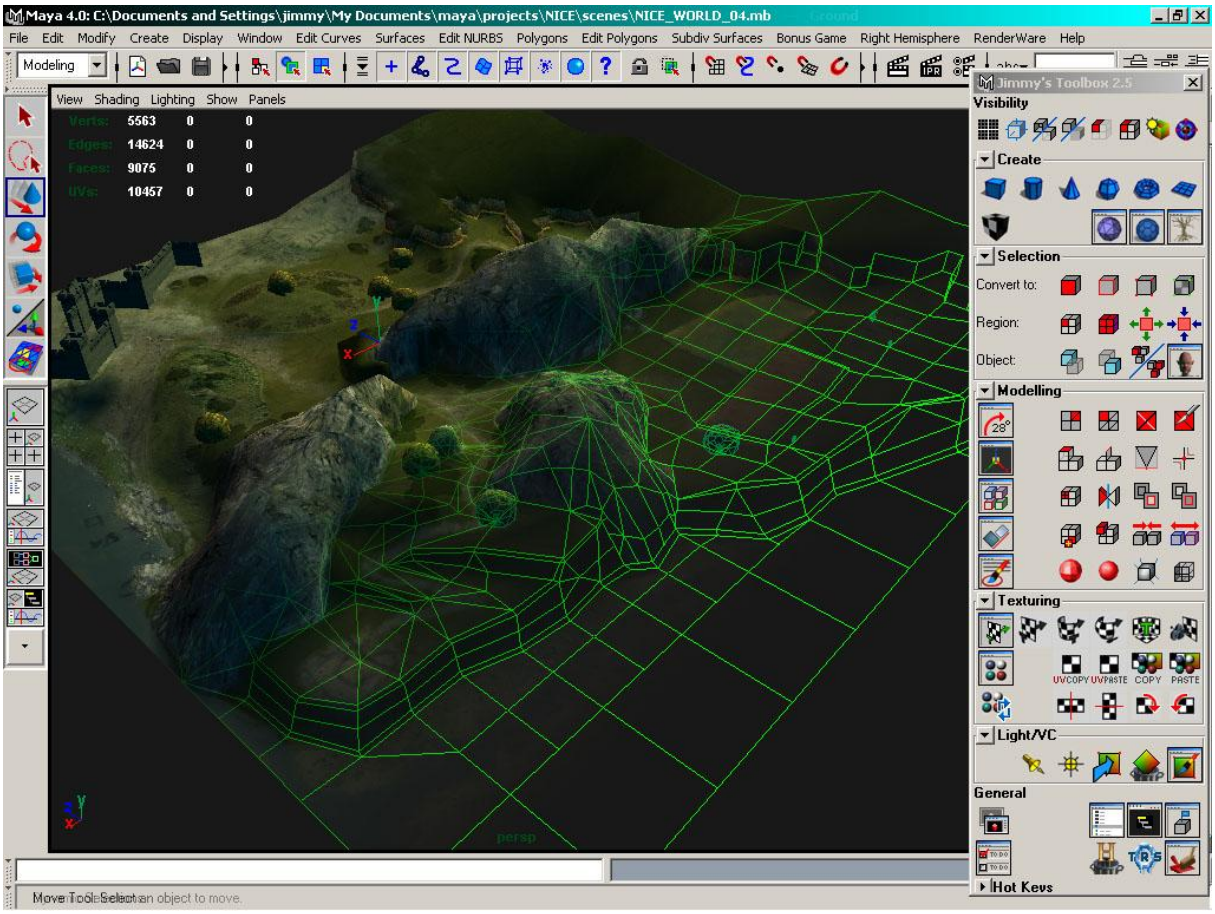

Figure 2.3.1. A 3-D model of the fairy-tale world.

Figure 2.3.2. A 3-D model of the HCA study.

2.4 Description of each module of the simulation layer

2.4.1 Simulation server

- *Function* : Keeps track of the absolute world state including characters and objects communicating this information on changes and/or on queries:

2.4.2 Character response animation system

- *Function* : processing and generation of animation (receives actions and additional information from the character module and a time stamped soundtrack from the text to speech module it then produces a synchronized animation and soundtracks)
- *Input* : actions and a timestamped soundtrack
- *Internal data and models used by this module* :
- *Output* : synchronized animation and soundtracks
- *Exemple of graphical response produced by this module* :

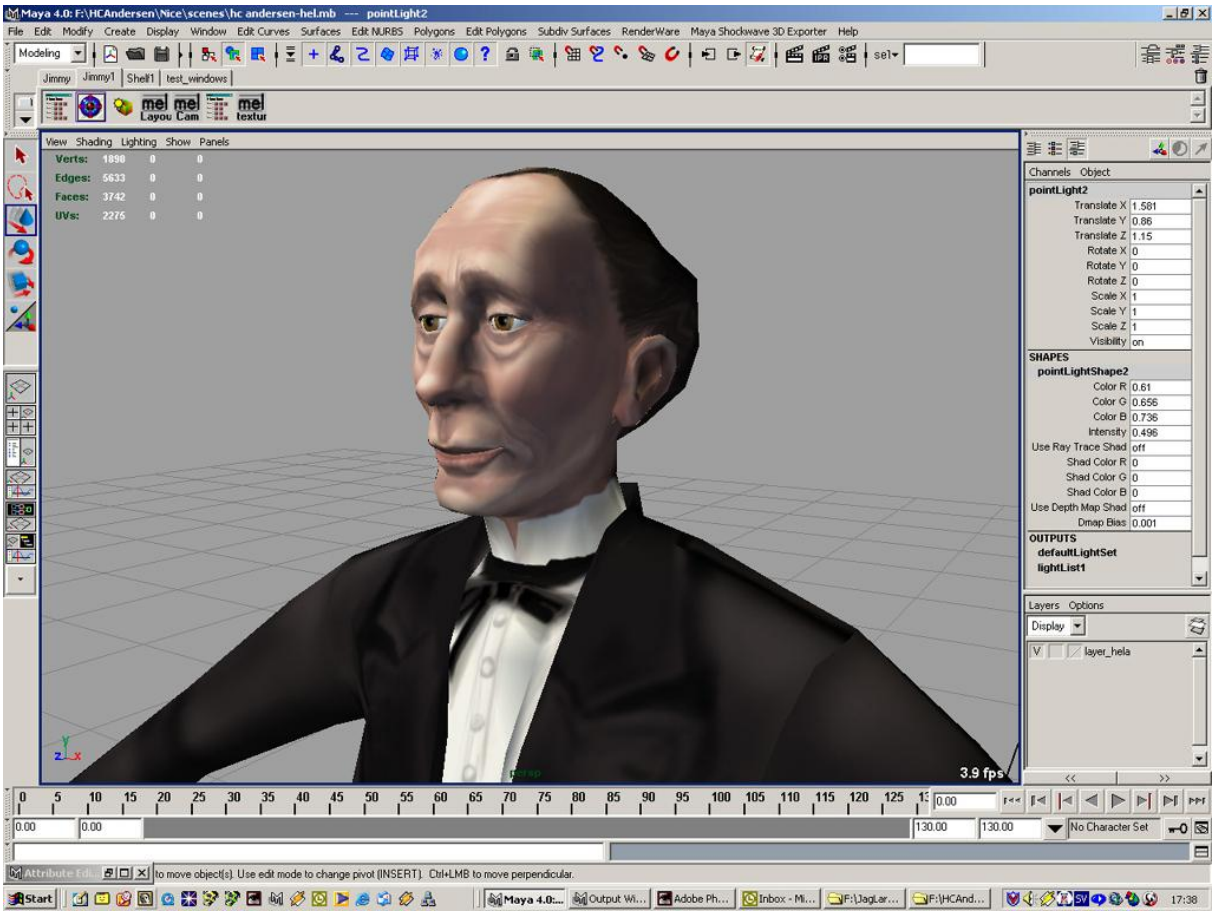

Figure 2.4.1. 3-D model of HCA.

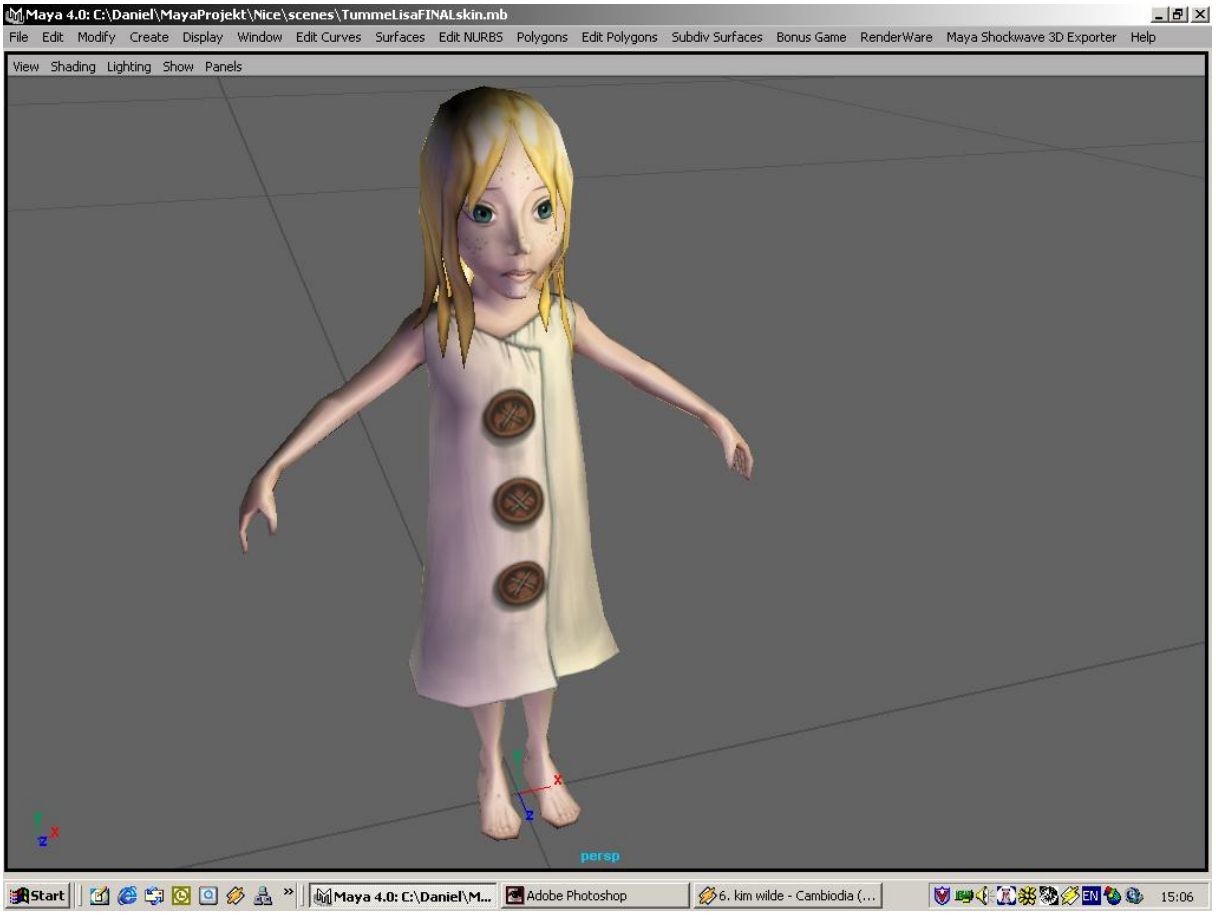

Figure 2.4.2. Thumbelina.

2.4.3 Text to speech

Input will be an XML-expression containing both animation tags, words to be synthesized and state tags (like emotion). The latter tags can be used in the production of the synthesized speech, e.g.

```
<emph level="0.5">
<emotion state="angry" level="0.9"/>
```

The animation tags specify animations that should be produced simultaneously with the synthesized speech, e.g.

```
<pointAt with="rightHand" at="chair01"/>
<eyebrowRaise with="both" amount="0.8"/>
```

The Speech Synthesis module will produce a sound file or audio stream to be sent to the Presentation Layer. The module will also compute timing information for the lip synchronisation, in the form of a list of the names of phonemes/visemes that occur in the utterance, along with their respective start times and/or end times. This list is sent to the Character Response Animation System together with the name of the sound file/audio stream. The Speech Synthesis module will also insert time stamps into the animation tags to facilitate synchronisation of speech and gestures. These time stamps are also user by the Character Response Animation System when producing the animations.

2.5 Description of each module of the character module layer

2.5.1 Gesture Interpreter

- *Function* : interprets gesture by merging results of gesture shape recognition and graphical objects location and types
- *Input* : recognized gestures + hypotheses on potential graphical referents
- *Internal data and models used by this module* : models of relations between gestures, graphical objects and commands as observed in Wizard of Oz studies (there might be a single interpreter with two different data model: one for HCA study and one for the fairy tale world)
- *Output* : possible interpretations of gestures and possible graphical referents

2.5.2 NLU for the fairy-tale part of the system

- *Function* : The NLU module (or parser) performs context-free interpretation of the user's utterances, producing semantic expressions as the result. The NLU module to be used in the fairy-tale part of the system is robust in the sense that it will always produce some result (i.e. all input is considered to be grammatical).
- *Input* : An n-best list and/or lattice of hypotheses from the speech recogniser.
- *Internal data and models used by this module* : The NLU module to be used in the fairy-tale world relies on several sources:
    - a *domain model* encoding the semantic functions that can constitute the interpretation of the user's utterance. Each semantic function has a number of typed arguments. This domain model is shared with the Input Fusion (IF) module.
    - a lexicon listing all the words that has a specific (non-null) interpretation in the domain.
- *Output* : An n-best list of semantic expressions encoding the possible interpretations of the user's utterance. This n-best-list will be XML-encoded.
- *Example of message produced by this module* : Consider the utterance "Where is Thumbelina?". It is a wh-question asking for a location, and could be represented in Prolog-style as follows: `wh(Location:_x, [thumbelina.pos=_x])` (variables are prefixed with underscore). Couched in XML it would look as follows:

```
<semanticRepresentation>
  <function>wh</function>
  <object>
    <type>Location</type>
    <value>_x</value>
  </object>
 <constraints>
    <eq>
      thumbelina.pos
      _x
    </eq>
  </constraints>
</semanticRepresentation>
```

2.5.3 NLU for the HCA part of the system

- *Function*: The function of the English NLU module is spot the topic being addressed by the user, identify speech act(s) and deictic expressions, resolve anaphora, and extract meaning.

- *Input*: An n-best list and a word hypothesis graph from the speech recogniser, information on changes to the scene on the screen provided by the simulation server.
- *Internal data and models used by this module*: At this point, we are still evaluating different types of parsing mechanisms suitable for the linguistic complexity present in conversational interaction between HCA and the users. One of our key priorities is to ensure fast processing of the input. Therefore, the exact data format and models used by the parser are still to be defined. We expect to include in the NLU a topic spotting mechanism which outputs the current topic of discussion based on recogniser input and information about topics previously addressed in the dialogue (if any). Modules for simple speech acts identification (three to five speech acts) and anaphora resolution will also be integrated in the NLU but are not yet specified in detail. Identification of deictic and possibly other expressions important to gesture identification in the Input Fusion module will be done as well.
- *Output*: An n-best list of semantic feature/value structures encoding the possible interpretations of the user's utterance. The information about changes to the scene on the screen. This output will be XML-encoded.
- *Example of message produced by this module*: In the case of the user saying "What is written here?" and assuming that the current topic of discussion is *Pictures_in_study*, we would get the output shown below. The *Time* attribute may be useful at the Input Fusion stage for appropriate pairings of gesture and deictic expression (like "here"). Each output is given a degree of confidence (*ConfidenceScore*). The *Domain* and *Topic* of the conversation are shown (the Domain attribute having been inferred from the Topic attribute). The Underspecified Semantics (*UnderspecifiedSemantics*) of the input, including anaphora, to be further resolved by the Input fusion and Dialogue Manager modules is computed, along with any deictic expression (*Deictic*) and speech act (*SpeechAct*).

```
<Spoken_Input>
<Time> 10.05:22 </Time>
<NLU>
        <ConfidenceScore> .900 </ConfidenceScore>
        <Domain> HCA_physical_presence </Domain>
        <Topic> pictures_in_study </Topic>
        <UnderspecifiedSemantics> written(Deictic)</UnderspecifiedSemantics>
        <Deictic> here </Deictic>
        <SpeechAct> question </SpeechAct>
</NLU>
</Spoken_Input>
```

2.5.4 Input Fusion

- *Function* : merge hypotheses from speech recognition and understanding with hypotheses from gestures interpretation (this "late fusion" might be extended latter in the project by earlier interactions between gesture and speech if it appears to be appropriate)
- *Input* : hypotheses produced by Natural Language Understanding and Gesture Interpretation module
- *Internal data and models used by this module* : models of cooperation between modalities as observed in Wizard of Oz studies + general rules regarding multimodal reference resolution

- *Output* : multimodal semantic hypotheses (those sent by NLU possibly completed by gesture semantics and possibly reordered) ; unsolved references might be solved by dialogue manager

2.5.5 Dialogue manager for the fairy tale world

- *Function* : The dialogue manager (DM) performs (dialogue-)context dependent interpretation of the user's input (resolving references, adding presupposed information, interpreting ellipses, etc.). The DM further updates the character's internal goal agenda, and possibly generates a speech act expression as a response.
- *Input* : An n-best-list of semantic expressions from the Input Fusion module.
- *Internal data and models used by this module* : The DM uses several sources:
    - the *agenda*, which encodes the long-term goals of the character.
    - the *dialogue history*, which encodes the past interactions between the user and the character
    - the *world model*, which models the character's knowledge about its environment.
- *Output* : Possibly a speech act expression to the Response Generator.

2.5.6 Dialogue manager for the HCA study

- *Function*: this module is the core module in the Character Module Layer because it has to always make sense of whatever input it receives from the Input Fusion module. As regards internal structure, the HCA dialogue manager (DM) will be significantly different from the DMs for the fairy tale characters.
- *Input*: N-best multimodal input semantics, information on changes to the scene on the screen.
- *Internal data and models used by this module*: The module will include the following functions: a Task Manager which controls the overall message passing in the module; a Non-Communicative Action generator which controls HCA's behaviour when he is not communicating with users; a fast-track Communicative Function module which controls HCA's behaviour when he is receiving the user's input; a Dialogue History which keeps track of the spoken dialogue history as well as of relevant facts about HCA in his virtual world setting, thus providing context for input interpretation; and a Mind State Agent which processes the input in its discourse context based on HCA's discourse plans, emotional state, knowledge as represented in the HCA Knowledge Base, and domain-based reasoning, and which eventually produces a semantic output representation for Response Generation. Pending final decisions the DM may work on a frame structure which eventually will hold the semantics to be passed on to the response generator. For fast-track input processing by the Communicative Function module a simpler data structure may be used. For querying the HCA knowledge base, XML-based queries will be used.
- *Output*: the DM sends the semantics of the output, including linguistics, actions, and emotions to the Response Generation module possibly in terms of a frame structure and wrapped in XML.

2.5.7 Response generation for the fairy tale world

- *Function* : The Response Generator produces a surface form expressing a speech act in words and gestures.
- *Input* : A speech act expression from the Dialogue Manager (DM).
- *Internal data and models used by this module* : Parts of the domain lexicon as well as rules to generate grammatical phrases and communicative gestures.

- *Output* : An XML-coded message specifying the character's response utterance and gestures.
- *Example of message produced by this module* : The following XML-expression encodes encodes a character saying "Do you want me to take this chair and put it here?", while pointing at the chair and the target position at the appropriate places in the utterance, emphasizing the word "me", sounding angry up until the word "chair", and then sounding surprised.

```
<animation type="sequential" id="25">
  <emotion state="angry" level="0.9"/>
  Do you want <emph level="1">me</emph> to take
  <emph level="0.5">
    <animation type="parallel">
      <animation type="sequential">
        <lookAt at="chair01"/>
        <pointAt with="rightHand" at="chair01"/>
        <lookAt at="user"/>
      </animation>
      this chair
    </animation>
  </emph>
  <emotion state="surprised" level="0.5"/>
  and put it
  <animation type="parallel">
    <pointAt with="leftHand" at="table02"/>
    <eyebrowRaise with="both" amount="0.5"/>
    here
  </animation>
</animation>
```

2.5.8 Response generation for the HCA study

- *Function*: The Response Generator produces a surface form of what is to be spoken to the user and of what is supposed to happen on the screen in terms of gestures and emotions being expressed.
- *Input*: Probably a frame structure from the DM expressing the output semantics.
- *Internal data and models used by this module* : Parts of the domain lexicon as well as rules to generate grammatical phrases and communicative gestures.
- *Output*: An XML-encoded message specifying the character's utterance, gestures and emotions, or only gestures if HCA is not talking to a user.
- *Example of message produced by this module*: The output will follow the same structure and XML-encoding, drawing on HCA relevant tags, as the fairy tale Response Generator. Thus we simply refer to the example provided for the fairy tale Response Generator.

3 Input

In order to develop multimodal systems, it is necessary to collect data on the behavior than can be expected from the future users (Martin, Julia et al. 1998). This section describes three preliminary experiments that have been achieved during the first year of the project. The goal of these experiments was twofold:

- collect data on the spoken and multimodal behavior that we can expect from the future users of the NICE system ;
- have experiences in collecting data in the case of adults/children multimodal conversation with virtual characters.

The results of these experiments will be used in preparing a more realistic data collection system which uses the 3D environment and 3D full body characters. Thus, data collected via these experiments are crucial sources of information for the development of the first NICE prototype (e.g. methodological recommendations, language models, experimental plateformes).

3.1 NISLab data collection on verbal conversations with a 2D talking head of Andersen

In order to collect acoustic and conversational data from interactions between typical users (children and adolescents aged 9 to 18) and an animated simulation of Hans Christian Andersen (HCA), Wizard of Oz experiments were conducted by NISLab with subjects using the set-up illustrated below.

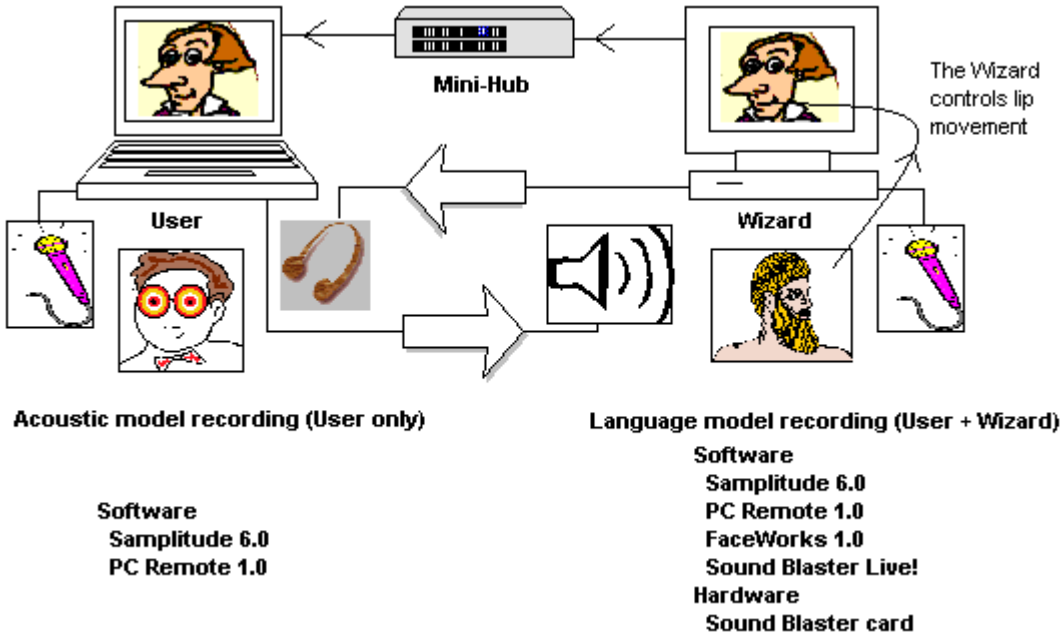

Figure 3.1.1. The Wizard of Oz set-up (NISLab).

To start the interaction with HCA, users are asked to state their age, gender and mother tongue. They speak through a microphone and listen to HCA's responses via headphones, allowing recording of the user's speech only. These recordings will be used to train the NICE speech recogniser for English. Moreover, the dialogues are being analysed to collect

information about what users actually want to talk to HCA about, and how they do this. The wizard speaks through a microphone and listens to the user via loudspeakers. Software voice distortion technology is used to render the voice of the wizard as neutral and computer-like as possible. The wizard is also responsible for controlling the lip movement of a cartoon-like HCA face, which can be viewed via a mini-hub connection by the user. The software used in the set-up is listed in Figure 3.1.1. Not illustrated is an analogue system for backup recording. To date, the system was successfully used to record around seven hours of conversation. Transcription of the recorded material has been done.

Average utterance length : 7 words

The exact number of user inputs collected is 2047 sentences distributed as follows:

- 01.10.02: 385 utterances
- 24.10.02: 569 utterances
- 28.10.02: 165 utterances
- 30.10.02: 510 utterances
- 31.10.02: 418 utterances

Recording conditions were :

- Microphone brand: ME2 Sennheiser
- Microphone Transducer principle condenser
- Microphone Sensitivity 20mV/Pa
- Microphone Sound pressure 130 dB SPL
- Microphone Pick-up pattern omni-directional
- Recording mode: 16 bits mono
- Recording sample rate: 32kHz

3.2 LIMSI-CNRS data collection on multimodal input behavior when interacting with 2D cartoon like characters

In order to collect data on multimodal behavior of the future users of the 3D NICE system, LIMSI-CNRS has carried out Wizard of Oz experiments with a simple game application including 2D cartoon-like characters which were designed using XML and Java technology. The multimodal (speech and pen gestures) behaviour of 7 adults and 10 children was video-taped and annotated. Behavioural metrics were extracted from the videos. These metrics, as well as subjective variables collected by means of a questionnaire, were then submitted to uni- and multidimensional statistical analyses.

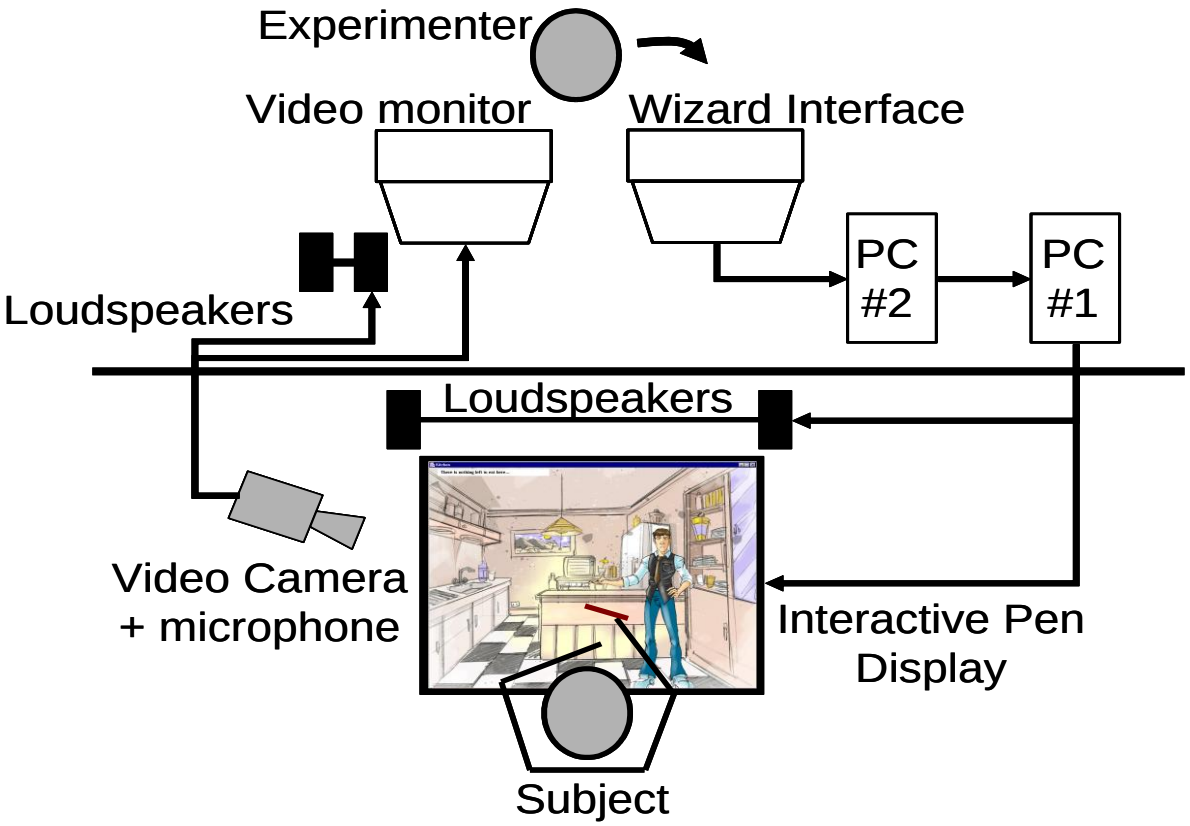

Figure 3.2.1. The Wizard of Oz set-up at LIMSI-CNRS for recording multimodal input behaviour from adults and children during interaction with 2D characters in a simple game application.

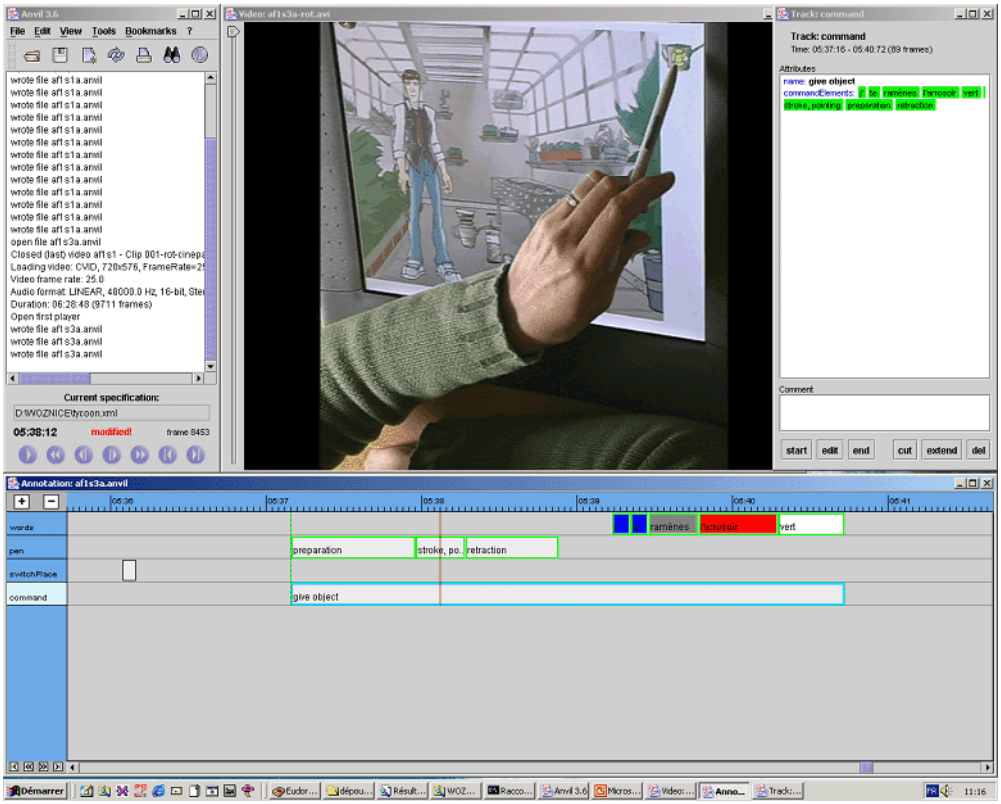

Figure 3.2.2: Frame of the annotation of a video showing pen and speech behavior when interacting with 2D characters and objects.

The 34 recorded videos (two scenarios for each of the 17 subjects) were then annotated. Speech annotations (segmentation of the sound-wave into words) were done with PRAAT[1] and then imported into ANVIL (Kipp 2001) in which all complementary annotations were made. Three tracks are defined in our ANVIL coding scheme:

- Speech, every word is labelled according to its morpho-syntactic category;
- Pen gestures (including the three phases: preparation, stroke and retraction) are labelled according to the shape of the movement: pointing, circling, drawing of a line, drawing of an arrow, and exploration (movement of the pen in the graphical environment without touching the screen);
- Commands corresponding to the subjects' actions (made by speech and/or pen). Five commands were observed in the videos: get into a room, get out of a room, ask a wish, take an object, give an object. Annotation of a command covers the duration of the corresponding annotations implied in the two modalities and is bound to these annotations.

Annotations were then parsed by Java software we developed in order to extract metrics that were submitted to statistical analyses with SPSS[2].

The results confirm the usefulness of multimodal input, which yielded shorter scenarios, higher and more homogeneous ratings of easiness. Additional results underlined the importance of gesture interaction for children, and showed a modality specialization for certain actions. Finally, multidimensional analyses revealed links between behavioral and subjective data, such as an association of pen use and pleasantness for children. These results

[1] http://www.fon.hum.uva.nl/praat/

[2] http://www.spss.com/

can be used for both developing the functional prototype and in the general framework of ECA-systems evaluation and specification.

3.3 Telia data collection of children's speech

Since the beginning of the NICE project, Telia has at its disposal a set-up in the Telecommunications museum in Stockholm for eliciting and collecting computer-directed speech from museum visitors. The set-up provides prompts generated by an animated speaking agent (see Gustafson and Sjölander 2002). The set-up has been put to use for gathering data for training of a first version of acoustic models of children's speech. So far (in December 2002), some 20.000 utterances have been recorded and transcribed, 40 % of which come from users up to 15 years of age. This data has been sent to ScanSoft for the training of Swedish acoustic models.

3.4 Telia data collection using a 2D virtual Cloddy Hans

A second Swedish data collection was carried out as a Wizard-of-Oz simulation in the Telecommunications museum. The purpose of this experiment was to answer two questions: First of all, how do people react when they are put in a situation where they are encouraged to interact with a virtual agent in collaborating and solving a simple task? Secondly, will the users adapt their way of speaking (as concerns the choice of words and speaking tempo) to the agent's way of speaking? The answer to these questions are of great interest when designing the fairy-tale part of the system.

Sixteen volunteer subjects, 9 men and 7 women between the ages of 17 and 59, participated in a WoZ experiment. The study was performed in the exhibition area of the Telecom museum in Stockholm. Most of the subjects were members of the general public, and some were museum employees. They were informed about the general purpose of the study, and were told that they were being recorded. Subsequently, subjects were given a pictorial scenario and instructions on how to talk into the microphone, and were told to await further instructions. The subjects got a prerecorded instruction on how to perform the actual task. This involved helping a story tale character, Cloddy Hans, to solve a puzzle. Subjects were told that the solution to the puzzle would be revealed to them after a number of colored geometrical figures had been moved from one part of the screen to another in a certain order. Since Cloddy Hans lacked information required to perform this task, the subjects had to help him using their pictorial scenarios. Cloddy Hans was not animated but he could be placed in one of five fixed poses indicating that he was talking, thinking, not understanding and so on.

Figure 3.4.1. A subject interacting with Cloddy Hans.

The total number of recorded user utterances was 297. Each dialogue consisted of 6 tasks (i.e. there were 6 colored geometrical figures to be moved) and 16 to 27 user turns. The total number of words in the corpus was 2173. The entire corpus was orthographically transcribed, and all user turns were tagged with information on position in the dialogue, type of user turn, previous system output, etc. At the word level, the user utterances were also labeled for lexical content, distinguishing between 'color', 'shape' and 'other' words. However, for the purpose of this particular study, only a subset of the user turns was of interest, namely the first turns in each task, and repetitions and rephrases of these. Thus, 130 user turns consisting of short comments such as "ok" or "yes" and a few erroneous utterances were excluded from the subsequent analyses. The average length of the remaining 167 utterances was about 6 seconds.

In short, the results of the study showed that users are enthrilled to interact with a virtual agent of their own size, and they did their best to solve the tasks that were put before them. The lexical entrainment effect was very strong, whereas users adapted their speaking tempo to a lesser degree.

4 Informal specification of multimodal output

The NICE embodied animated interface agents will behave in ways which may best be characterised as an approximation to human natural interactive communication and natural human action more generally. The distinguished fiftyish H. C. Andersen (HCA) will exhibit rather "brainy" conversational communicative behaviour and relatively little physical action whereas the fairy tale characters will produce a mixture of game physical action and, compared to HCA, simpler communicative behaviour. The domains of knowledge and discourse of HCA will be his life, his fairy tales, his physical setting and presence, his role as "gate-keeper" to the fairy tale world which is accessible through a door in his study, and the information he gathers about individual users during conversation. The domains of knowledge, discourse, and action of the fairy tale characters are being specified at the time of writing.

In general, the output produced by HCA and the fairy tale characters will be (a) speech and (b) graphical output behaviours. The spoken output is presently being specified for HCA by NISLab and for the fairy tale characters by Telia, and will not be discussed further in this deliverable. However, to ensure the production of increasingly powerful graphically rendered output behaviours for HCA and the fairy tale characters through a series of versions of the NICE graphical rendering system, the first one of which will be the data collection system due in March 2003, NICE partners NISLab and LIMSI have conducted a series of literature studies, data collections, and brainstorming exercises in order to identify the graphical elements which Liquid Media will need in order to render the actions and communicative behaviours needed in NICE. The results of these studies are presented in Sections 4.1 and 4.2 below, followed by a presentation by Telia of markup language proposals to be used by the NICE character modules in order to make the rendering system execute the graphical behaviours needed at any point in the conversation or game.

The primary sources for NISLab's input to the definition of NICE graphical behavioural elements (Section 4.1) have been brainstorming based on two hours of video recording of Denmark's only full-time HCA imitator. The video recordings are available on CD-ROMs as a NICE development data resource. The primary sources for LIMSI's input to the definition of NICE graphical behavioural elements (Section 4.2) have been literature studies and the LEA embodied animated interface agent. In both cases, the results arrived at are presented in a series of tables which should be interpreted as follows.

We have first looked at (a) emotions, attitudes, cognitive stances, and communicative actions which we would like to have realised in the NICE embodied animated interface agents. For all of these, the selection criterion has been that they are likely to occur so frequently during agent communication and action in interaction with NICE system users that it is important and worthwhile to render them. In the tables in Sections 4.1 and 4.2, these emotions, etc. are typically being presented in the left-most column. Secondly (b), we have defined a series of graphical behavioural element *types*, such as "posture" or "eye shape", and tried to characterise each emotion, attitude, cognitive stance, or communicative action in terms of specific behavioural elements of those types. This methodology has been very useful for structuring our brainstorming. Eventually, however, what we will end up with is a consolidated specification structure consisting of (i) the graphical behavioural element *types* which have been agreed upon by NISLab, LIMSI, Telia, and Liquid Media, and, for each type, a list of *specific* behavioural elements which will be available for the NICE output designers. Armed with this structure, the NICE output designer will be able to express an open-ended series of emotions, attitudes, cognitive stances, communicative actions, and physical actions only subject to the major criterion that each of them should be graphically

expressed in a realistic and natural fashion. Correspondingly, the animation designers will have been satisfied that the types-cum-element structure aimed at will conform to the way in which graphical renderings are actually being produced, so that the implementation of the structure will be as straightforward to do as demanded by the behavioural elements required and agreed upon. A further advantage is that the behavioural element type-cum-specific element structure will be applicable to *any* NICE embodied animated interface agent.

Time-wise, we expect to have arrived at a consolidated specification structure agreed upon by NISLab, Telia, LIMSI, and Liquid Media by early May 2003. This structure will obviously still be preliminary because the ongoing specification of HCA's spoken conversation and the fairy tale characters' speech-cum-action will no doubt require additions to the structure. Still, the structure will enable Liquid Media to implement a broad selection of behavioural elements for HCA and the fairy tale characters, which will enable us to collect increasingly realistic data resources on children's interactions with the NICE characters.

4.1 Suggestions grounded on observations from video of HCA behavior

The NICE HCA actor video was recorded in February 2003 by Andrea Corradini and Svend Kiilerich, NISLab, in collaboration with the HCA imitator and actor Torben Iversen. The actor teaches afternoon drama classes with Danish children. The video was recorded with two of the actor's classes with children aged around nine and twelve, respectively. At a preparatory meeting with the actor, it was agreed that, prior to the recording with the children, he would spend an hour with the NISLab video team acting out HCA's non-communicative actions (NCAs), i.e. the actions which HCA will perform on-screen when he is not engaged in conversations with users. Following these recordings, the drama class students would be individually instructed to ask specific questions to HCA to which he would then respond, everything being recorded on video. The questions were intended to be representative of the NICE HCA's domains of knowledge and discourse. In addition, the children were asked to provide free-style questions and other conversational input to HCA, all of which was recorded as well. The recorded videos were analysed and the analysis formed the basis for the behavioural elements analysis presented below.

Figure 4.1.1. HCA writing.

Figure 4.1.2. HCA is happy about what the user just said.

This section proposes a first specification of the graphical behavioural elements involved in HCA's communicative actions (CAs), non-communicative actions (NCA's), and communicative functions (CFs). *Communicative action behaviour* is the character's behaviour while responding to the user's communication as made through speech and gesture. *Non-communicative action behaviour* is the character's behaviour whilst not engaged in conversation with a user. *Communicative function behaviour* is the character's behaviour whilst the user is talking and/or gesturing.

4.1.1 Communicative Actions

4.1.1.1 Emotions

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Anger	Sitting or Standing	Look at user	Front	Thumping the desk with the fist while sitting.	Eyebrows frowned	
Happiness	Sitting or Standing	Look at user	Front	Arms Open	Smiling	Smile broadly
Sadness	Sitting or Standing	Look down			Sad	
Normal (Friendly state)	Sitting, legs crossed.	Look at user	Front	One hand on the table. The other resting on the leg.	Friendly	
Normal (Friendly state)	Sitting, (Body leaning forward towards the user)	Look at user	Front	Both arms resting on the legs.	Friendly	
Normal (Friendly state)	Standing	Look at user	Front		Friendly	

The Normal (friendly state) has three different behaviours because HCA would spend a lot of time in the normal (friendly state). The state has to have a rich set of behaviours that can be randomly selected.

4.1.1.2 Other Emotions

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Pride	Sitting or Standing	Look far away	Up		Eyebrows raised	Closed or smile
Shock/ Surprise	Sitting or Standing	Look at user	Front	Arms open	Eyes wide open	Wide open

Note: - Emotions under this category have lower priority than the emotions in the first table. These could be implemented in the first prototype, if possible, or in the second prototype.

4.1.1.3 Turn Taking

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Giving Turn	Sitting or Standing	Look at user	Front	Arms crossing	Depends on emotion	Closing
Taking turn	Sitting or Standing	Look at the side, then at user	Turn away then front	Moving hands/arms a bit	Depends on emotion	Opening

4.1.1.4 Emphasis

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
	Sitting or Standing. In both cases the upper body would lean forward	Look at the user	Front	Finger pointing towards the user	Wide open	

4.1.1.5 Greetings

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Welcoming the User	Turn to user if needed.	Look at user	Lift head if necessary	Welcoming user gesture	Smiling	Smile
Saying Goodbye	*Would remain in the same posture	Look at user	Short nod	Wave goodbye	Depends on emotion	Depends on emotion

* HCA would not change his posture otherwise when the user is leaving or when he welcomes the user. For example, if HCA is sitting when the user arrives, he will keep on sitting and smile at the user, he would not change his posture. The same would apply when he says goodbye to the user.

4.1.1.6 Other Behaviours

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Writing in the Air	Sitting or Standing	Front	Front	One arm will do the action of writing in the air.	Depends on emotion	Depends on emotion
Disagreement	Sitting or Standing	Look at user	Shaking his head	Arms crossed	Depends on emotion	Depends on emotion

Agreement	Sitting or Standing	Look at user	Nodding		Depends on emotion	Depends on emotion
Writing with the pen	Sitting	Look at the paper	Looking down	One Arm holding the pen and writing, the other resting the table	Friendly	Mumbling.
Point to Object	Sitting or Standing	Towards the object	Front	Point with the finger towards the object	Depends on emotion	Depends on emotion
Hold Object	Sitting or Standing	Look at object then at user	Front	Hands around object		
Pick Object	Sitting or Standing	Towards the object.	Front	Hands on the object		
Put down the object	Sitting or Standing	Towards the object	Front	Hands on the object		
Show Object	Sitting or Standing	Towards the user or towards the object.	Head position will follow accordingly	Hands on the object.	Depends on emotion	Depends on emotion
Turn around	Standing	Away from user, then towards user.	Head position will follow accordingly.			
Put hat on	Sitting or Standing	Towards the user	Front	Hands on the hat	Depends on emotion	Depends on emotion
Put hat off	Sitting or Standing	Looking sideways	Sideways.	Hands holding the hat.	Depends on emotion	Depends on emotion

4.1.2 Non-Communicative Actions

	Posture	**Gaze**	**Head Position**	**Arm Position**	**Eye Shape**	**Mouth Shape**
Write	Sitting	Looking towards the paper	Looking down	One Arm holding the pen and writing, the other resting on the table.		
Walk Around in Study	Standing	Look in front	Front	Clasp hands back or Put hands in pocket.		
Look at Books on the book	Standing	Look at book	Front	Fold arms		

shelf						
Sit in Easy Chair Thinking	Sitting	Look sideways	Up	Scratch Head.		
Sit in Easy Chair reading	Sitting	Look at the book.	Front	Hands holding the book		

4.1.3 Communicative Functions

Elements and comments -> Output modalities	**Behavioural elements**	**Comments**
Speech	yes, right, okay, ehmm, well, no_speech	Add more elements?
Gaze	look at user, look away, look down, stare, roll eyes	Note that funny looks are possible, such as rolling the eyes
Gesture		Study what people do!
Facial expression	smile friendly	The CF only produces this one (default) facial expression
Head movement	Nod	Add more elements?
Physical action	suspend action, dismantle action	Seems to cover the possibilities at this point
Body posture	should follow the above, as appropriate	Taken care of by the RE

4.2 Suggestions grounded on the LEA cartoon-like agents

This section provide some suggestions mostly based on experiments with the LEA 2D cartoon-like agents. It provides references and informal proposals for multimodal behaviors in the following sections: emotional behaviors, communicative acts, speech turn behavior, and other behaviors.

The following tables present some internal states or intentions and the corresponding animation of several modalities, as reported in the literature.

Each internal state can be displayed in several ways. It can be expressed sometimes by an animation in a single modality. For example, a smile can be enough to express happiness. In this respect, asterisks have been added in the tables to indicate animations which seem to be sufficient to express alone the corresponding state. Other modalities can be animated simultaneously to reinforce this expression by redundancy or complementarity.

However, co-occurrence of several behaviours could happen to involve contradictory animations. This kind of conflict can be solved by using alternative animations for a single modality (dashes inside the same cell) and/or animations of other modalities. For example, the agent can have to talk and show happiness at the same time. Although s/he can't smile while talking, happiness could be shown by talking fast, opening the mouth relatively wide, and/or by displaying screwed up eyes.

These multiple ways of displaying internal states may also provide diversity.

All the animations extracted from these tables could be displayed simultaneously. A few exceptions are indicated by a specific remark.

4.2.1 Emotional behaviors

Examples can be found in (Blair 1995, Poggi et al. 2000; VHML 2001, Williams 2001).

* have been added in the tables to indicate animations which seem to be sufficient to express alone the corresponding state

State of the agent	Head position	Eyes shape	Gaze direction	Mouth shape	Speech flow	Arm gestures
Happiness	Front	- Screwed up ** - Eyebrows raised	- Middle - Up	- Smile *** - Wide open	Fast	- Large gestures - Hands up - Arms opened
Sadness	Down	Inner eyebrows up and central ***	Down	- Corners down ** - Closed	Slow	- Shoulders down ** - Arms dangling - Symmetrical movements
Surprise	Front	Eyebrows raised, eyes wide open ***	Middle	Wide open	Fast	- Arms open - Hands up
Fear		Eyebrows raised, eyes wide open ***		Wide open		
Uncertainty, powerlessness	Front	Eyebrows raised		Corners down	Slow	- Shoulders shrug *** - Hands down - Hands opened
Certainty	Front	Eyebrows frowned **	Middle			
Pride	Slightly up	Eyebrows raised, eyes half-closed ***			Normal / slow	Hands on the waist
Anger		Eyebrows frowned ***	Middle	Closed	Fast	Hands on the waist (+ stamping feet)

State of the agent	Head position	Eyes shape	Gaze direction	Mouth shape	Speech flow	Arm gestures
Tiredness	Down	Half-closed **	Down	- Yawning ** - Half-open	Slow	- Symmetrical and slow movements - Hand on the mouth while yawning
Intimidation	Down ***	Eyebrows raised	Up	Half-open	Slow	- Hands behind the back - Hands down, put together
Impatience	Up	- Eyebrows raised - Eyebrows frowned	Up	Closed	Fast	Arms folded ** (+ stamping feet)
Thinking	Up	- Eyebrows raised - Eyebrows low	- Up - Away	Closed	Slow	- Hand on the chin ** - Scratching the head **

4.2.2 Communicative acts

Examples can be found in (Pelachaud et al. 1996, Johnson et al. 2000).

Communicative act	**Head position**	**Eyes shape**	**Gaze direction**	**Mouth shape**	**Speech flow**	**Arm gestures**
Reacting to another agent	Towards the agent ***		Towards the agent ***		Interruption	
Salutation	Head nod **	Eyebrows raised	Middle	Smile		Waving **
Reacting to someone else's action	Towards the object manipulated ***		Towards the object manipulated ***			
Asking a question	Front	Eyebrows raised **	Middle		Normal	
Agreement	Head nod ***		Long blinking	- Closed - Smile		
Disapproval	Shaking **	Closed				
Auto-designation	Front		Middle			Hand / finger on the breast **
Emphasis	Head nod	- Eyebrows raised ** - Voluntary blink				
New topic of conversation						Change of body posture and orientation
Pause		Voluntary blink *				
End of sentence	Slow head movement coming to rest *					

4.2.3 Speech turn behaviors

Examples can be found in (Cassell et al. 2000, Cassell & Vilhjalmsson 1999).

Speech turn behavior	**Head position**	**Eyes shape**	**Gaze direction**	**Mouth shape**	**Speech flow**	**Arm gestures**
Giving speech turn	Front **	Eyebrows raised	Middle **	Closed	Silence ***	Relaxing hands
Asking speech turn	Front **		Sustained glance **	- Half-open - Smile	Silence	Raising hands **
Taking speech turn	Left or right		Away **		Normal	Starting gesturing **
Requesting feedback	Front **	Eyebrows raised **	Middle **		Silence	
Giving feedback	Head nod **		Middle	Closed	- Silence *** - "Hmmm"	
Breaking away from conversation	Left or right		Away ***	Closed		

4.2.4 Other behaviors

Examples can be found in (Blair 1995, Gustafson 2002, Johnson et al. 2000, Williams 2001).

Behaviors	**Head position**	**Eyes shape**	**Gaze direction**	**Mouth shape**	**Speech flow**	**Arm gestures**
Showing something	Turned towards the object **		Towards the object **			Pointing with the hand / with the finger anywhere ***
	Animation of head and gaze can occur *before* arm gesture.					
Reference to small objects		Half-open, eyebrows low				Hands close to each other ***
Reference to big objects		Wide open, eyebrows raised				Hands outspread ***
Offering something	Front	Eyebrows raised	Middle	Smile		Carrying an object and holding it to an agent / the user ***
Taking something	Towards the object		Towards the object			Catching the object ***
	Animation of head and gaze can occur *before* arm gesture.					

4.3 Mark-up language

4.3.1 Introduction

This section outlines a suggestion for a language for describing multimodal output from the character modules. The formalism is heavily inspired by VHML (Gustavsson, Strindkund and Wiknertz 2001, Marriot and Stallo 2002, Beard and Reid 2002), AML (Kshirsagar et al 2002, Arafa et al 2002), CML (Arafa et al 2002), Alice (Conway et al 2000), XSTEP (Huang, Eliens and Visser 2002) and other systems and formalisms.

In what follows, we will assume that we have defined a set of basic atomic animated actions. An "atomic action" will be an action that can be considered atomic from the point of view of the behaviour system of a character, like raising an eyebrow or picking up an object. Note that these atomic actions might correspond to very sophisticated animation procedures in the simulation layer. For instance, picking up an object must be rendered differently depending on how the character is positioned relative to the object, whether the object is on the ground, on a table or on a high shelf, whether the object is big or small, heavy or light, etc.

4.3.2 Basic actions

Basic actions will be coded as empty-element XML tags, e.g.

```
<lookAt at="chair01"/>
```

Every kind of action has its set of attributes which are relevant for that action, some of which are compulsory while others are optional. For instance, it seems reasonable that `lookAt` has a compulsory attribute `at`, and several optional attributes (like `duration`, that specifies how long the animation should last).

All optional attributes should have a default value. A good idea from Alice is that the animation of any action should last 1 second, unless otherwise specified (and unless other animation constraints forces the animation of that action to be be longer or shorter, see below). Another idea from Alice is that each range should be between 0 and 1 (e.g. how much do you want to raise the eyebrow, turn the torso, etc.), where 1 is maximum and 0 is minimum.

To make a complete animation instruction, we will enclose the action in a pair of animation tags (the reason for this will be clear below), as follows:

```
<animation>
  <lookAt at="chair01"/>
</animation>
```

4.3.3 Combining actions

In order to create complex actions, we will want to combine basic actions, either by animating them in sequence or in parallel.

If we want our character to first look at the chair, then point at it using the right hand, and then look back at the user again, this is achieved by the command:

```
<animation type="sequential">
  <lookAt at="chair01"/>
  <pointAt with="rightHand" at="chair01"/>
  <lookAt at="user"/>
</animation>
```

To point at the table with the left hand and simultaneously raise the eyebrows, the following command can be used:

```
<animation type="parallel">
  <pointAt with="leftHand" at="table02"/>
  <eyebrowRaise with="both" amount="0.8"/>
</animation>
```

Not all actions can be done in parallel, notably combinations that defy the laws of nature (like looking right and looking left at the same time, or sitting down while walking, etc.). But also other, more reasonable combinations can be extremely hard to combine in a natural-looking way, like picking up an object while walking past it, etc. So a good rule-of-thumb is that any two actions that affect the same joint in the bone model of the character cannot be done in parallel. (Of course, this might sometimes be somewhat difficult to predict for the behaviour system of a character. But probably most combinations we will want to do will be possible, like combining facial expressions with arm gestures).

Parallel and sequential actions can be arbitrarily nested. For instance, to first look at the chair, then point at it while simultaneously raising the eyebrows, write:

```
<animation type="sequential">
  <animation>
    <lookAt at="chair01"/>
  </animation>
  <animation type="parallel">
    <pointAt with="rightHand" at="chair01"/>
    <eyebrowRaise with="both" amount="0.8"/>
  </animation>
</animation>
```

4.3.4 Combining speech and animations

A string to be synthesized by the speech synthesizer should also be looked at as an animation, since the lip movements of the character will be synchronized with the speech. So to make the character say "Hello there", we should send the following expression to the simulation layer:

```
<animation type="sequential">
  hello there
</animation>
```

Of course such utterances can be combined with other animations, e.g.:

```
<animation type=”parallel”>
  <raiseArm with=”rightArm”/>
  <animation type=”sequential”>
    hello there
  </animation>
</animation>
```

would result in the character saying “hello there” while simultaneously raising the right arm. In order to simplify the syntax somewhat, we can allow ourselves to skip the `animation` tags around the words:

```
<animation type=”parallel”>
  <raiseArm with=”rightArm”/>
  hello there
</animation>
```

Emphasis is marked the standard way by enclosing the words (and actions) with `emph` tags:

```
<animation type=”sequential”>
  do you mean
  <emph>
    <animation type=”parallel”>
      <pointAt at=”chair01” with=”rightHand”/>
      this
    </animation>
  </emph>
</animation>
```

4.3.5 Emotions

Emotions like angry, sad, surprised, etc., are signalled either by an empty-element tag, e.g.

```
<emotion state=”angry” level=”0.9”/>
```

which switches the mood of the character, or by pair of tags:

```
<emotion state=”angry” level=”0.9”> … </emotion>
```

which sets the mood of the character to angry for the time it takes to render whatever is between the start and the end tag (and then the character goes back to whatever mood it had before).

4.3.6 Computing animation timings

It was stated above that each animation should last 1 second unless otherwise specified. When speech is involved, however, the time it takes to utter the words should be the guiding quantity. So if the character is to say "hello there" while simultaneously raising his arm, and it takes 0.75 seconds to say "hello there", then obviously the arm should also be raised in 0.75 seconds. Such timings are calculated by the Character Response Animation System in the Simulation Layer That module will communicate with the speech synthesizer to obtain the times needed to utter the words, as well as with the Simulation System in order to get times needed to perform certain physical actions (like walking from A to B).

The XML representation will be split into two parts: one for the lip synchronisation and one for the gesture and action animation. The text-to-speech system will be used to generate the list of phonemes and their respectively timings. This list will be sent separately to the Character Response Animation System for lip synchronization. In our example, it may look like this:

```
<animation type="sequential">
  <emph time="1300">
    <animation type="parallel">
      <pointAt at="chair01" with="rightHand" startTime="1300" endTime="1690"/>
    </animation>
  </emph time="1690">
</animation>
```

The Character Response Animation System will produce an action and gesture animation track using the timing constraints, and this will then be fusioned with the lip-synchronisation track to produce the final animation track that will be sent to the rendering system.

4.3.7 A complicated example

The following expression is probably as complicated (or more complicated) as any expression we will use. It encodes a character saying "Do you want me to take this chair and put it here?", while pointing at the chair and the target position at the appropriate places in the utterance, emphasizing the word "me", sounding angry up until the word "chair", and then sounding surprised.

```
<animation type="sequential" id="25">
  <emotion state="angry" level="0.9"/>
  Do you want <emph level="1">me</emph> to take
  <emph level="0.5">
    <animation type="parallel">
      <animation type="sequential">
        <lookAt at="chair01"/>
        <pointAt with="rightHand" at="chair01"/>
        <lookAt at="user"/>
      </animation>
      this chair
    </animation>
  </emph>
```

```
  <emotion state="surprised" level="0.5"/>
  and put it
  <animation type="parallel">
    <pointAt with="leftHand" at="table02"/>
    <eyebrowRaise with="both" amount="0.5"/>
    here
  </animation>
</animation>
```

4.3.8 The message flow within the system

We assume that expressions like the ones discussed above will be generated by the Character Modules and sent to the Message Dispatcher (MD). The MD will send the expression to the Character Response Animation System, which will compute the timings of the animation, contact the Speech Synthesizer to produce a sound track, while simultaneously producing an animation track. These parallel tracks will finally be sent to the output terminals, where they will be rendered.

After the utterance and/or movements have been produced on the output side, a message is sent back to the character module.

5 Future directions

This report has presented the NICE architecture as specified in terms of main modules and main module communication. Barring the odd question of detail to be resolved among the partners, we are now at the point where detailed main module specification can take place in the knowledge that the main system architecture has been agreed upon. We are continuing discussion of the *input devices* which users will use to interact with the NICE system. Even if we know that they will communicate with the NICE using the interactive modalities of speech and 2D gesture, these constraints are compatible with a wide selection of input devices. For instance, the NICE system wants to know when there is a new user or if the previous user has left, but this knowledge can be retrieved from a push-to-talk button, a user sitting down in a chair in front of the system, a user uttering the right (word-spotted) words to a constantly open speech recogniser, a user gesturing without speaking, or ..., etc. Moreover, even if we have agreed that the user will have a first-person perspective in the NICE virtual world, we need further investigation to determine by means of which input devices the user will move around in the virtual world, will address-through-gesture objects and people, etc. Current thoughts address (a) putting the user in a chair in front of the NICE system to ensure unambiguity of user presence and –change, effective microphone placement for speech recognition, and the opportunity to have the microphone and speech recogniser open at all times; and (b) soft(ware) avatar motion and pointing devices in order to free the user from having to operate several different manual (hardware) devices. However, the different NICE settings currently experimented with include full-figure back-projection, home settings, large flat screen setting in which the combination of a chair and touch screen use may be awkward, etc. So, we continue to discuss the NICE interactive setting(s) we want to aim for. Nevertheless, these discussions on peripherals are not likely to seriously interfere with our preparations of the core modules for the first NICE prototype.

As for NICE system output, we have, in a sense, come farther than as regards system input. NICE system output will be rendered through text-to-speech and through on-screen graphics. We are close to consortium-wide agreement on the graphical output elements which we shall have for the first NICE prototype (Section 4). Still, the issues involved in the currently discussed NICE interactive output setting options, such as full body-sized projection, touch screen. large screen, standard PC screen, remain unresolved.

Another important point is the following which concerns both the to-be-expected NICE system user input and the NICE speech-graphics output. We are keenly aware that, to proceed beyond draft *systematic* domain, communication, and action specification, such as the ongoing specification of HCA's twenty or so main fairy tales out of a total of around 170 fairy tales, or of HCA's life as a particular instance of a human life in general as viewed by that human himself, and to proceed beyond current (by its nature, non-systematic) *use case-based* analysis of the processing of user input through the entire NICE system, we need to collect far more data on contemporary children's conceptions of HCA, his fairy tale characters, and games involving those fairy tale characters. Without rich information on contemporary children's conceptions of the NICE system, we will not succeed in building a NICE application which is sufficiently entertaining, in addition to being educational, for our core user group which is children aged 8 to 18 years old. NISLab's data gathering has so far yielded five hours of English children's conversation with HCA. This is a considerable resource the transcription of which we continue to study. However, the children were constantly prompted through graphical (text) means with inspirational topics of conversation with HCA. In the absence of those, heavily priming, promptings, we still do not know what

today's children want to talk to HCA about. The same comments apply to the fairy tale world games with HCA's fairy tale characters: which games will today's children want to play, what will be the entertaining contents of those games, what can spoken dialogue contribute to computer gaming in particular and in general, etc.? To find out more on this crucial issue, we urgently need a non-priming first NICE data collection system.

6 References

AAMAS 2002. Workshop on "Embodied conversational agents - let's specify and evaluate them!", Marriot, A., Pelachaud, C., Rist, T., Ruttkay, S., Vilhjalmsson, H. (Eds.) pp 42-28. http://www.vhml.org/workshops/AAMAS/papers.html in conjunction with The First International Joint Conference on Autonomous Agents & Multi-Agent Systems, 16 July, 2002, Bologna, Italy

Allbeck, J. & Badler, N. (2002). Toward Representing Agent Behaviors Modified by Personality and Emotion. Workshop on "Embodied conversational agents - let's specify and evaluate them!", in conjunction with The First International Joint Conference on "Autonomous Agents & Multi-Agent Systems", Bologna, Italy.

Arafa et al (2002) " Two approaches to Scripting Character Animation", http://www.vhml.org/workshops/AAMAS/papers/Kamyab.pdf

Arafa, Y., Kamyab, K., Mamdani, E., Kshirsagar, S., Magnenat-Thalmann, N., Guye-Vuillème, A. & Thalmann, D. (2002). Two approaches to Scripting Character Animation. Workshop on "Embodied conversational agents - let's specify and evaluate them!", in conjunction with The First International Joint Conference on "Autonomous Agents & Multi-Agent Systems", Bologna, Italy.

Beard and Reid (2002) "MetaFace and VHML: A First Implementation of the Virtual Human Markup Language" http://www.vhml.org/workshops/AAMAS/papers/beard.pdf

Blair, P. (1995). Cartoon animation. Walter Foster Pub.

Buisine, S., Abrilian, S., Rendu, C. & Martin, J.-C. (2002). Towards Experimental Specification and Evaluation of Lifelike Multimodal Behavior. Workshop on "Embodied conversational agents - let's specify and evaluate them!", in conjunction with The First International Joint Conference on "Autonomous Agents & Multi-Agent Systems", Bologna, Italy.

Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (2000). Embodied conversational agents. MIT Press.

Cassell, J., Vilhjalmsson, H. (1999). Fully embodied conversational avatars: making communicative behaviors autonomous. Autonomous Agents and Multi-Agent Systems, 2, 45-64.

Conway, M. et al (2000) "Alice: Lessons Learned from Building a 3D System for Novices", http://www.alice.org/publications/pubs/chialice.pdf

Daghstul seminar on "Coordination and Fusion in Multimodal Interaction" 29 October 2001 - 2 November 2001 http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/

Gustafson, J and Sjölander, K (2002) "Voice Transformations For Improving Children's Speech Recognition In A Publicly Available Dialogue System", Proceedings of ICSLP02, Colorado USA

Gustafson, J. (2002). Developing multimodal spoken dialogue systems. Doctoral dissertation.

Gustavsson, Strindkund and Wiknertz (2001) "VHML draft", http://www.vhml.org/downloads/VHML/vhml.pdf

HF 2002. Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges 29th November, 2002 Melbourne, Australia http://www.vhml.org/workshops/HF2002/papers.shtml

Huang, Eliens and Visser (2002) "XSTEP: A Markup Language for Embodied Agents", http://www.cs.vu.nl/~eliens/projects/@archive/refs/xstep2.pdf

Johnson, W.L., Rickel, J.W., Lester, J.C. (2000). Animated pedagogical agents: face-to-face interaction in interaction learning environments. International Journal of Artificial Intelligence in Education.
Johnson, W.L., Rickel, J.W., Lester, J.C. (2000). Animated pedagogical agents: face-to-face interaction in interaction learning environments. International Journal of Artificial Intelligence in Education.
Kipp, M. (2001) Anvil - A Generic Annotation Tool for Multimodal Dialogue. In Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech), pp. 1367-1370, Aalborg, September 2001.

Kshirsagar et al (2002) "Avatar Markup Language", http://www.miralab.unige.ch/IgnasiCFTEST/DynamicSite/3research/assdb/papers/96.pdf

Marriot and Stallo (2002) "VHML - Uncertainties and Problems. A discussion…" http://www.vhml.org/workshops/AAMAS/papers/marriott.pdf

Martin, J. C., Julia, L. & Cheyer, A. (1998). A Theoretical Framework for Multimodal User Studies. Conference on Cooperative Multimodal Communication, Theory and Applications (CMC'98), Tilburg, The Netherlands.

Pelachaud, C., Badler, N.I., Steedman, M. (1996). Generating facial expression for speech. Cognitive Science, 20(1): 1-46.

Pirker, H. & Krenn, B. (2002). Deliverable D9c of the NECA project: Report on the assessment of existing markup languages for avatars, multimedia and multimodal systems on the WWW., OFAI.http://www.ai.univie.ac.at/NECA/publications/publication_docs/d9c.pdf

Piwek, P., Krenn, B., Schröder, M., Grice, M., Baumann, S. & Pirker, H. (2002). RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA. Workshop on "Embodied conversational agents - let's specify and evaluate them!", in conjunction with The First International Joint Conference on "Autonomous Agents & Multi-Agent Systems", Bologna, Italy.

Poggi, I., Pelachaud, C., De Rosis, F. (2000). Eye Communication in a Conversational 3D Synthetic Agent. In: E. Andrè, (Ed.), Special Issue of Artificial Intelligence Communications, The European Journal on Artificial Intelligence, IOS Press, vol 13 (3): 169-181.

PRICAI 2002. International Workshop on Lifelike Animated Agents Tools, Affective Functions, and Applications held in conjunction with Seventh Pacific Rim International Conference on Artificial Intelligence August 19, 2002 Tokyo, Japan. http://www.miv.t.u-tokyo.ac.jp/~helmut/pricai02-agents-ws.html

VHML (2001). Working draft on VHML specification.

Wahlster, W. (2001). SmartKom - A Transportable and Extensible Multimodal Dialogue System. Seminar on "Coordination and Fusion in Multimodal Interaction", Daghstul.

Wahlster, W., Reithinger, N. & Blocher, A. (2001). Smartkom: Multimodal Communication with a Life-Like Character. EuroSpeech'2001, Aalborg (Denmark).

WG4-Daghstul (2001). Working Group on "multimodal meaning representation".http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/

Williams, R. (2001). The animator's survival kit. Faber & Faber.