NICE project (IST-2001-35293)

Natural Interactive Communication for Edutainment

NICE Deliverable D5.1a

First Prototype Version of Conversation Management and Response Planning for H.C. Andersen

15 October 2003

Authors

Niels Ole Bernsen and Laila Dybkjær

NISLab, Odense, Denmark

Project ref. no.	IST-2001-35293
Project acronym	NICE
Deliverable status	Restricted
Contractual date of delivery	1 September 2003
Actual date of delivery	15 October 2003
Deliverable number	D5.1a
Deliverable title	First Prototype Version of Conversation Management and Response Planning for H.C. Andersen
Nature	Report
Status & version	Final
Number of pages	20
WP contributing to the deliverable	WP5
WP / Task responsible	TeliaSonera
Editor	Niels Ole Bernsen
Author(s)	Niels Ole Bernsen and Laila Dybkjær
EC Project Officer	Mats Ljungqvist
Keywords	Conversation management, response planning, life-like animated characters
Abstract (for dissemination)	This report, Deliverable 5.1a from the HLT project Natural Interactive Communication for Edutainment (NICE), describes conversation management and response planning for life-like animated conversational agent H. C. Andersen in the first NICE prototype.

Table of Contents

1 Introduction

This NICE Report D5.1a describes conversation management and response planning for Hans Christian Andersen in the first NICE prototype (PT1). PT1 includes two generic life-like animated character software modules, one for Andersen (HCA) and one for the fairy tale characters. This is due to the fact that HCA and the fairy tale characters play very different roles in the NICE system. Whereas HCA is "brainy" and conversational, and also rather static in a physical sense because he is confined to his study during interaction with users, the fairy tale characters are physically active interface agents with less conversational skills than HCA. The fairy tale character software module is described in D5.1b.

The WP5 description underlying this report emphasises the objective of character software re-use, stating that a software kernel will be developed which is the same for each character and which can accommodate implementation of different character profiles. We will only be developing one conversational character of the HCA type in NICE, i.e. HCA himself. However, it has been an important design goal to develop the HCA character in such a way that the kernel character software can be re-used for other conversational characters of the same generic type as HCA. Similarly, the WP5 description mentions the goal of being able to parameterise the character kernel by a body of domain knowledge and a personality trait specification described in a representation language to be developed. For the HCA character type, we have developed the kernel software structure in such a way that, by replacing HCA's body of domain knowledge with the domain knowledge of somebody else, and by replacing his personality traits by those of somebody else, the kernel can be re-used for representing other character profiles. We are aware, however, that more research is needed in order to make the HCA character module kernel as generic as possible in principle.

According to the WP5 description, the HCA kernel will implement the following computational steps:

1. resolution of discourse references, ellipses and deictic references when these have not been resolved by the input fusion module but require access to the dialogue history to be resolved;
2. dialogue act classification of user utterances for use in querying the characters' personalities and knowledge bases;
3. dialogue history for keeping track of the discourse context;
4. decision on the next communicative action(s) to be performed by the focal character, including meta-communication;
5. response planning.

In PT1, we have implemented Steps 3, 4 and 5 and the part of Step 1 which deals with deictic reference resolution relating to input fusion. For HCA, we currently expect to use input fusion only in the limited fraction of cases in which users indicate objects in his study which they want to talk to HCA about. Discourse references (Step 1) will be addressed in the development of PT2 based on more user behaviour data than we have at present. It remains to be seen if ellipses (Step 1) pose any particular problems which need be addressed. Robust and shallow parsers, such as the one used for parsing input to HCA, tend to be quite tolerant of discourse ellipsis. As for Step 2, it should be noted that the term 'dialogue acts' is being used in very different ways in the literature on spoken and multimodal dialogue. Since the natural language understanding module which processes input for HCA carries out some amount of syntactic parsing og highly diverse spoken input language, we do not need the term ‘dialogue act classification of user utterances’ for describing what the parser delivers. The parser delivers a semantic representation of the user’s utterance at a more fine-grained level than that of dialogue acts. On the other hand, we shall need ‘dialogue acts’, i.e. *speech acts* in the classical sense of the term, in PT2 in order to analyse user input to decide if the user does or does not take the initiative in a particular conversational turn. For this purpose, we will use a high-level and, indeed, classical taxonomy of speech acts.

The present report draws upon information provided in NICE reports D1.1 and D1.2a. D1.2a lists the main differences from D1.1 in HCA PT1 character module specification. In addition, D1.2a provides PT1 specifics on the representation of domain information and personality information for HCA, and describes the underlying design ideas behind the representation and coding of HCA’s conversation behaviour. This information is not repeated in the present report. Thus, major chunks of PT1 HCA conversation management and response planning have been described in D1.2a already. In the present report, we put it all together, present the information flow model adopted for HCA in PT1, and describe, in the process, modules and functionalities which have not been described in D1.2a.

For another important source of information for the present report, we would like to mention the data collection results obtained in WP2 so far. In preparation for PT1 development, NISLab has carried out two separate Wizard of Oz studies of simulated English conversation with young users. The first one was done in the field at schools in Odense in the autumn of 2002, collecting 6 hours of data which have been transcribed and analysed. The second was done in the field at the HCA Museum in Odense in the summer of 2003. It followed the HCA PT1 specification and 30 hours of spoken conversation data were collected. The transcription is almost completed at the time of writing. Analysis has begun, yielding several corrections to the PT1 specification, and this data will form an invaluable source of information for PT2 development. Figure 1.1 portraits HCA himself. Figures 1.2, 1.3 and 1.4 illustrate the Wizard of Oz setups for the first and second HCA Wizard of Oz simulations.

The present report has received important input from work in WP3 on multimodal output, including NICE Report D.3.3. The HCA output generation module is described in NICE Report D3.7. It has, obviously, been important to broker between the development of the HCA character module and the development of the HCA response generator in order to make sure that the character module can deliver to the specification of the response generator.

Finally, the present report is a result of dialogue within WP4 on animated character rendering, cf. NICE Report D4.2. In particular, the HCA behaviour and physical appearance design, including the design of his study, have been communicated to WP4 in order to ensure that HCA and his study are rendered in a reasonably historically correct fashion in PT1 and that HCA has the non-verbal behaviour repertoire needed for him to act quasi-naturally in PT1.

In what follows, Section 2 briefly describes what we consider to be the main challenge we have been facing during development of conversation management and response planning for HCA in NICE PT1, i.e. the spoken conversation challenge. Section 3 presents the PT1 HCA character module architecture. Section 4 describes the HCA character module as an finite-state output machine. Based on the context provided by these sections, Section 5 first describes the functionality and information flow of the HCA character module’s core input processing module, the mind state agent. The following sub-sections describe the functions of the character module manager, the conversational intention planner, the mind state agent manager, and the conversation history, respectively.

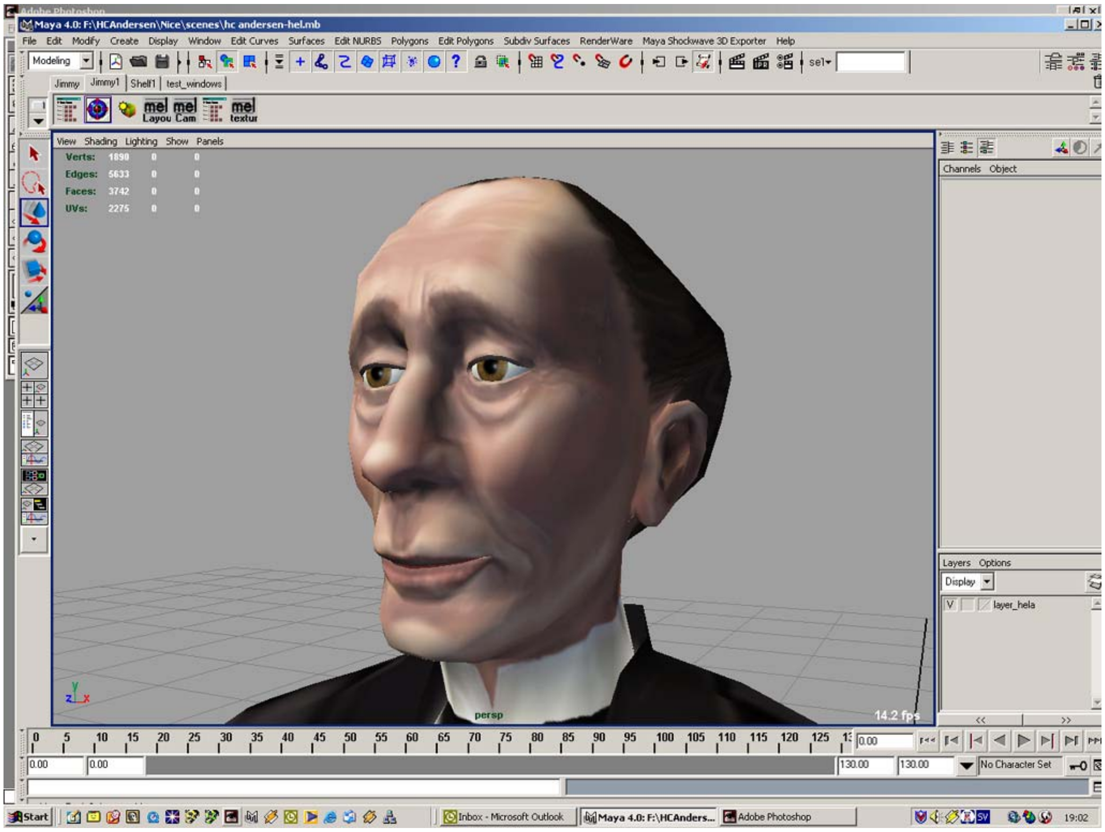

Figure 1.1. NICE fairy tale author Hans Christian Andersen.

Figure 1.2. Conversation with the wizard in the first NISLab Wizard of Oz simulation.

Figure 1.3. HCA Museum setup for the second NISLab Wizard of Oz simulation.

Figure 1.4. HCA and users in the second NISLab Wizard of Oz simulation.

2 The main challenge in PT1

There are several state-of-the-art research challenges involved in developing the HCA character module, including the 'kernel' challenge of building software which is easily re-usable for realising characters with knowledge and personalities very different from HCA's. However, we believe that the main challenge in PT1 is to enable HCA to successfully carry out *real* conversational dialogue. This merits some explanation. In recent years, the animated interface character community has been misusing the term "conversation" to such an extent that 76 command word applications have been described as conversational systems. This inflation in the use of the concept of conversation is no doubt due, in part, to the fact that most of our colleagues in the animated interface character community do not come from the spoken dialogue systems crowd. For the latter, the *non-conversational,* task-oriented spoken dialogue system remains the all-dominant paradigm, see, e.g., [Bernsen et al. 1998].

By contrast, HCA is not task-oriented at all. He is, rather, what we have chosen to call *domain-oriented,* i.e. he is able to conduct real conversation about several different domains without the – for the spoken dialogue systems designer – comforting and richly constraining limitations imposed by the inherent logic and combinatorics involved in carrying out dialogue about some particular task. HCA is, on the other hand, not able to conduct conversation about any domain whatsoever. If he were, he would be a contender in the competition for the first system to pass the Turing test. In other words, domain-oriented conversational dialogue is a half-way post between task-oriented dialogue and full Turing-compliant conversational abilities. Finally, the concept of conversation itself has an inherent richness which we try to take into account in developing HCA, a richness which goes completely against that of the notion of a task-oriented dialogue. The concept connotes, among other things, cf. D1.2a, the following principles for successful prototypical human-human conversation:

- initially, the interlocutors *search for common ground,* such as shared interests, shared knowledge, and similarity of character and personality, to be pursued in the conversation;
- the conversation is successful because the interlocutors *find enough common ground* to continue the conversation;
- the interlocutors provide, by and large, *symmetrical contributions* to the conversation, for instance by taking turns in acting as domain experts, so that one partner does not end up in the role of passive hearer/spectator, for instance like the novice who is being educated by the other(s);
- to a significant extent, the conversation is characterised by the participants taking turns in *telling stories,* such as anecdotes, descriptions of items within their domains of expertise, jokes, etc.;
- conversation is *rhapsodic,* i.e. highly tolerant to digression, the introduction of new topics before the current topic has been exhausted, etc.; and
- conversation, when successful, leaves the partners with a sense that it has been *worthwhile.*

The PT1 implications for HCA's dialogue behaviour are described in D1.2a.

3 Architecture

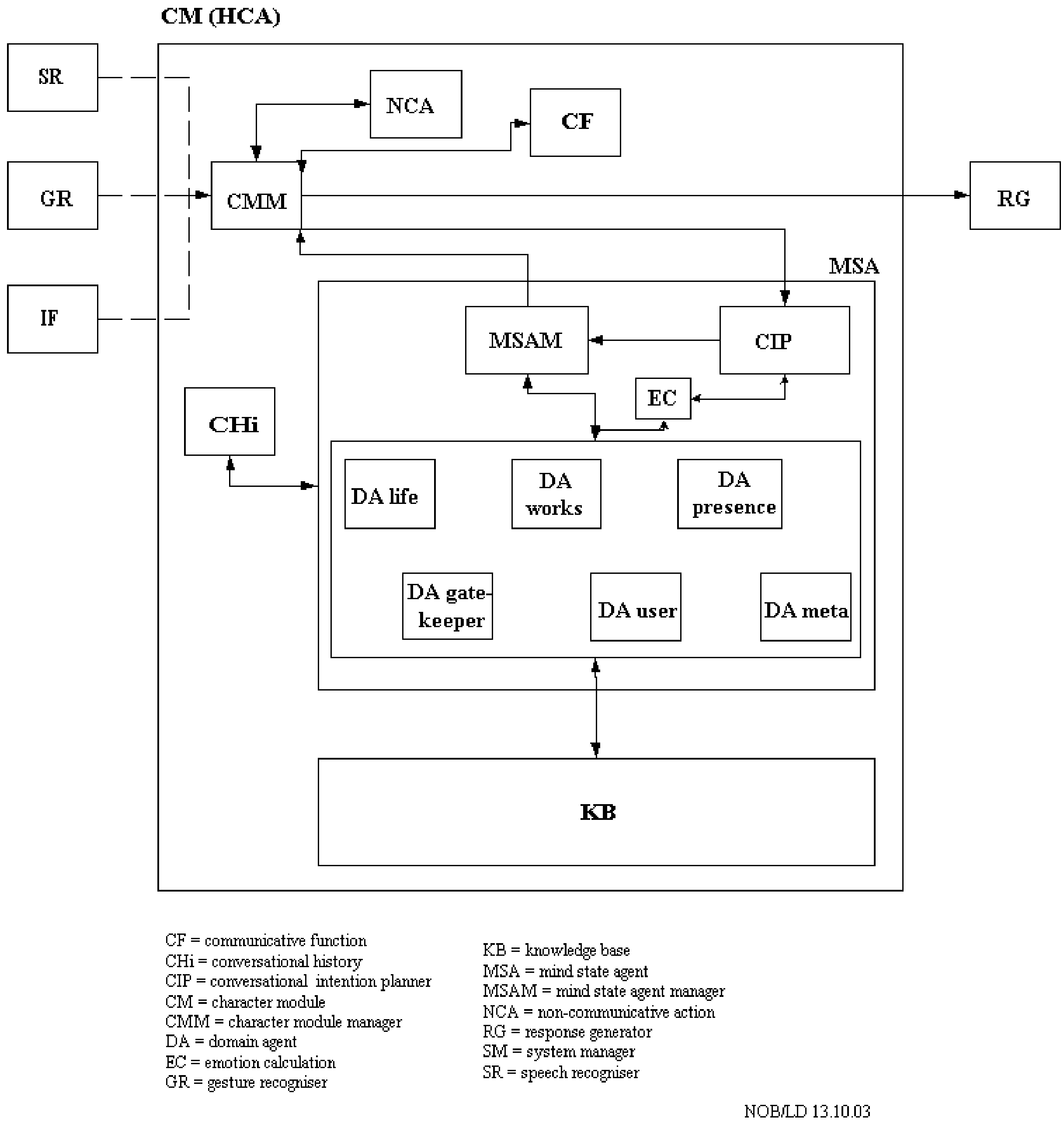

Figure 3.1. High-level architecture of the HCA character module.

The focus on domain-oriented conversation is reflected in the fact that we no longer speak of a dialogue manager but of a *character module* which conducts domain-oriented conversation, including response planning. In fact, also the term 'response planning' is obsolete in this context because what the character module does is conversational contribution planning. Moreover, the character is no longer part of a unimodal spoken dialogue system but is a life-like animated conversational character. An important implication is that we need to distinguish between three different situations as regards the conduct of conversation. In the first situation, the character is producing a conversational contribution using spoken and non-verbal communication. In the second, the character is receiving the user's spoken and 2D gesture conversational contribution. In the third, the character is visibly alone and has to behave meaningfully even if not engaged in conversation. We call the corresponding three system output states communicative action (CA), communicative function (CF), and non-communicative action (NCA), respectively.

Figure 3.1 shows the PT1 HCA character module architecture. The mind-state agent (MSA) does communicative action, the communicative function (CF) module takes care of communicative functions, and the non-communicative action (NCA) module takes care of non-communicative action. The character module manager (CMM) controls those modules and communicates with the following external modules: speech recogniser (SR), gesture recogniser (GR), input fusion (IF), and response generator (RG). In terms of the general NICE PT1 architecture, the HCA character module acts as a dialogue server, cf. Figures 3.2 and 3.3 from NICE report D3.7.

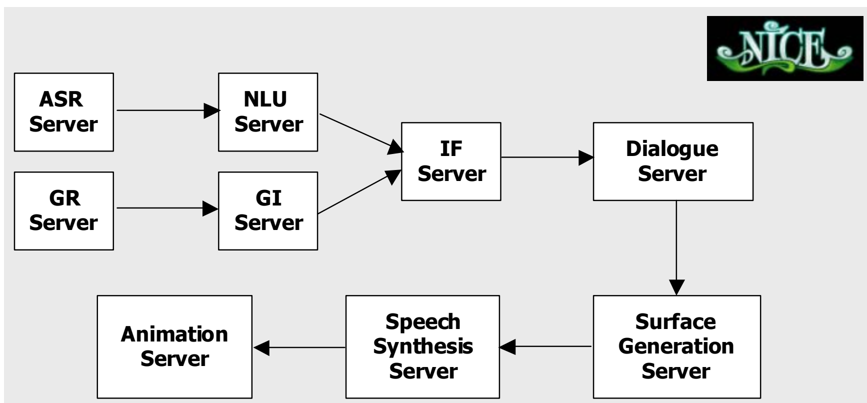

Figure 3.2. Information flow in the first prototype.

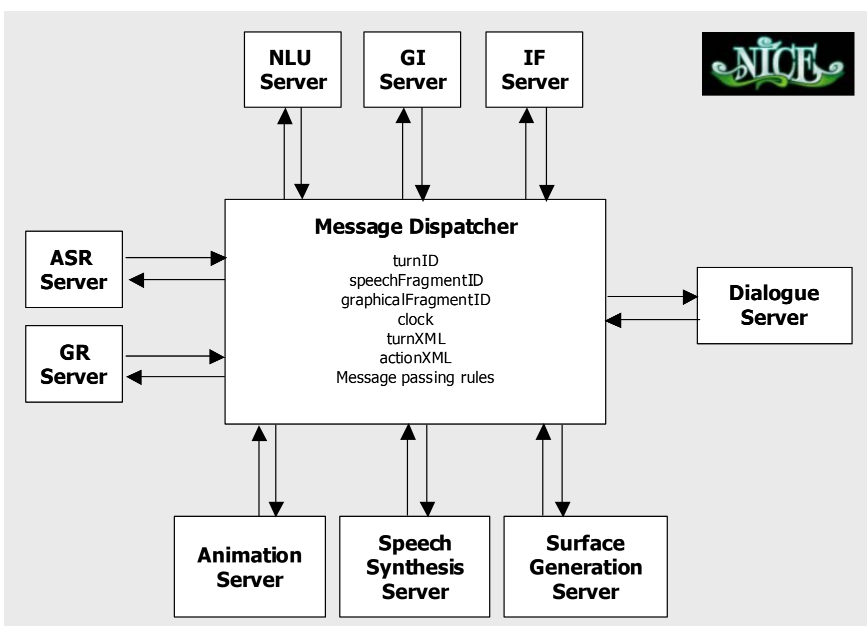

Figure 3.3. System architecture of the first prototype.

4 Output states, non-communicative action and communicative functions

Viewed as a whole, the HCA character module (CM) is a finite-state machine which is controlled by the character module manager (CMM, Figure 3.1) and which is in one of three output states, producing either:

- non-communicative action (NCA) output
- communicative function (CF) output
- communicative action (CA)

The overall state relationships are:

```
NCA -> CF <-> MSA -> NCA
```

Thus, NCA, the system's "resting state" must be followed by a CF state in which a new user starts addressing HCA. Following the user's first conversational contribution, conversation in which user and system take turns [CF <-> MSA] may go on for a while, eventually being followed by the NCA state.

A high-level state table is shown in Table 4.1.

Module/state	**Start/end state**	**Message**	**Condition(s)/notes**
NCA	start_processing	system_init	-
	start_processing	SR_timeout	-
	end_processing	*CF_output*	only when NCA is active, of course
CF	start_processing	SR_onset	-
	start_processing	GR_onset	-
	end_processing	*MSA_output*	-
MSA	start_processing	external input to the CMM	-
	end_processing	*NCA_output*	-

Table 4.1. NCA, CF, MSA state table draft.

Table 4.1 shows that, whilst in the CF state, the character module, i.e. its mind state agent (MSA), simultaneously processes the user's most recent input. Further design analysis demonstrated that the draft state table shown in Table 4.1 had to be extended in order to take into account a series of non-standard situations in which the user, for instance, does not respond within a fixed period of time. Figure 4.1 shows the normal interaction flow.

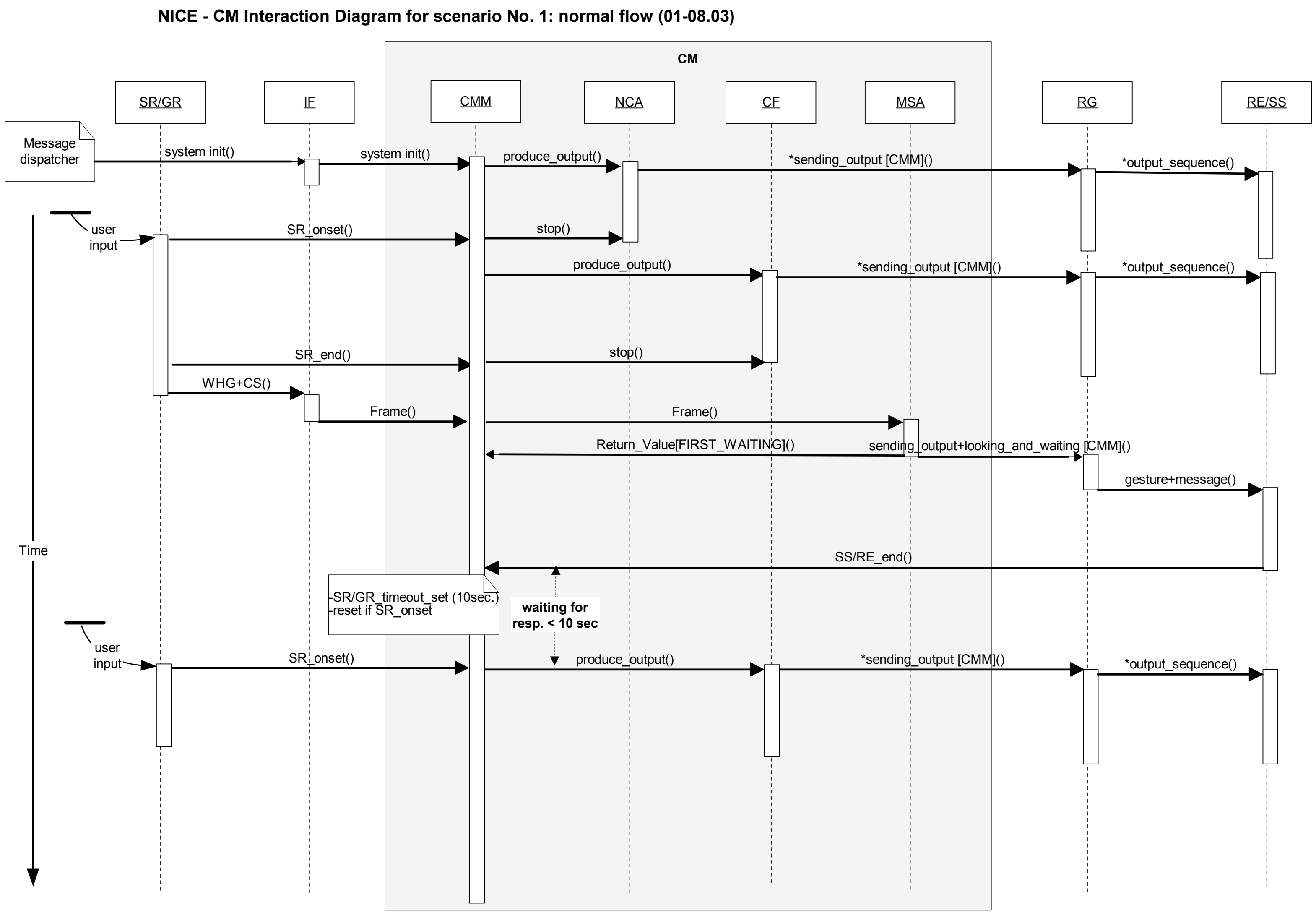

Figure 4.1. NCA, CF, MSA flow diagram, standard conversational interaction.

When in the non-communicative action output state, the CM is designed to produce "endlessly" varied (>one hour), looping spoken, non-verbal acoustic, and non-verbal behaviour as shown in Table 4.2, cf. [Bernsen 2003].

Behaviours **Output modalities**	**Behavioural elements and action sequences**
Speech	mumble unintelligibly, mumble words out of context
Other acoustic elements	hum a tune, grunt, hrmmm, action noises
Gaze	look down, look into space, roll eyes, stare, close one eye
Gesture	clasp hands front or back,, put hands in pockets, fold arms, one/two-arm gesture, wave arms, scratch head
Facial expression	look thoughtful, smile, set lips in various ways
Head movement	nod, shake head
Body posture	stretch body (sitting or standing)
Emotional background	friendly, at ease
Action sequences	write, walk around, look at books on shelf, sit in easy chair thinking or reading, look at wall pictures

Table 4.2. NCA repertoire: emotional background, action sequences, and behavioural elements [Bernsen 2003].

When in the communicative function output state, the CM is designed to produce spoken, non-verbal acoustic, and non-verbal behaviour as shown in Table 4.3, cf. [Bernsen 2003]. The particular behaviours to be shown are selected through weighted randomisation at run-time. An important point about communicative functions in today's systems is that the CFs cannot be made to reflect the user's current input because the speech recogniser only delivers its recognition results when recognition has been completed. It follows that the character's CF behaviours have to be somewhat "poker-faced" as the character cannot respond to, e.g., shocking news in real time.

Behaviours **Output modalities**	**Behavioural elements and action sequences**
Speech	yes, right, okay, ehmm, well, none
Gaze	look at user, look away, look down, stare, blink
Facial expression, body	determined by latest CA
Head movement	nod, none
Physical action	suspend NCA, dismantle NCA

Table 4.3. CF repertoire: behavioural elements and graceful NCA dismantlement [Bernsen 2003].

5 User input management and response planning

The mind-state agent (MSA, Figure 3.1) manages the user's spoken and/or gesture input including the planning of which response (or conversational contribution) to produce to the input. The MSA is controlled by the character module manager (CMM). In addition, the MSA communicates with the character's knowledge base (KB) and with the conversation history (CHi). Internally, the mind-state agent includes two sets of modules which jointly manage the user's input, i.e. the conversational intention planner (CIP) together with the emotion calculation (EC), and the six domain agents (DAs), one for each of HCA's six domains. The mind state agent manager (MSAM) mainly has a management role. The knowledge base (KB) is a database which includes references to all of HCA's coordinated spoken and non-verbal output. The conversation history (CHi) includes a comprehensive record per input and output turn of the current dialogue. All modules in the mind-state agent draw upon CHi information during input processing.

In D1.2a, we have described the personality representation functionality of the conversational intention planner and the representation of domain information in the knowledge base. In the following, we first describe the overall flow of information involving the mind-state agent. We then describe the modules which have not been described in D1.2a, i.e. the character module manager, the conversational intention planner's overall functionality, the mind state agent manager, and the conversation history.

5.1 Mind state agent information flow

The user's input is sent to the character module manager, CMM (Section 5.2) by the input fusion module. The CMM sends the input to the conversational intention planner. The user's input is represented in a frame. Slots in the frame which may be filled when the CMM receives the frame are shown in the Table 5.1 which also includes a column to the right exemplifying the contents of each slot.

Attribute	Values	Example
Ling. conf. score only	1, 2, 3 (highest), one per input turn	3
Gesture conf. score only	1, 2, 3 (highest), one per input turn	3
Ling.+gesture conf. score	1, 2, 3 (highest), one per input turn	3
Speech domain(s)	work, life, physical presence, user, gate-keeper, meta 1-best	physical presence
Speech topic(s)	1-best. See PT1 topics list	study
Speech semantics	1-best. See PT1 semantics list	what(object)
Speech act	question or statement; will not be used in PT1	-
Gesture semantics	gesture function(s), gesture object(s) 1-best	pointing, picture of Jenny Lind
Scene information	Values not known at present	HCA is standing at his writing desk

Table 5.1. CIP input.

Gesture and linguistics+gesture confidence scores are being post-processed by the character module to yield the values 1, 2 and 3 in Table 5.1. In Table 5.1, we have left out event messages from the speech recogniser and possibly from other modules, such as the input fusion module. The reasons are that we still do not know if the PT1 NICE HCA system will manage to include the Scansoft speech recogniser. This is not in the contract. Moreover, at the time of writing, there is still uncertainty with respect to the exact information which the HCA character module will receive from the input fusion module, cf. NICE Report D3.6.

The conversational intention planner (CIP) processes the input according to the character's conversational agenda and based on the character's personality (Section 5.3). The input is sent to the relevant domain agent via the mind state agent manager (Section 5.4). After CIP processing, the input frame may have been expanded with the slots shown in Table 5.2.

Attribute	Values	Example
Priority value set	[(domain, topic, index, speech act)*]	[(works, ugly duckling, 2, null)]
Emotional state	(happy, angry, sad)	(0, 0, 0)
Meta pattern length	integer	0
Action	enumeration type	null
Skip_replies	true or false	false

Table 5.2. Conversational intention planner -produced frame slot values.

In Table 5.2, the priority value set describes the continuation proposed by the CIP, cf. NICE Report D1.2a. Since the PT1 HCA system does not deal with speech acts, the speech act value will always be null. Emotional state functionality and computation is described in report D1.2a. The example emotional value in Table 5.2 is HCA's default emotional state. The action value, if different from null, indicates one of a number of meta-communication actions to be handled by the Meta DA. Skip_replies indicates whether HCA will provide an answer to the user's input before continuing with the output resulting from the priority value set.

When the CIP has finished processing, the mind state agent manager (MSAM) takes over, cf. Section 5.4. The MSAM communicates with the domain agents which are described in more detail in D1.2a. After MSAM processing, the input frame may have been expanded with the slots shown in Table 5.3.

Attribute	Values	Example
Response list	list with references to linguistic and behavioural output	[T1, T4]

Table 5.3. Slot values produced by the mind state agent manager.

The way in which a response list is built is described in Section 5.4. A response list contains references to one or more templates (T) each of which may contain linguistic, behavioural, or both kinds of elements. See also D3.7 for details on response generation.

5.2 Character module manager

The character module manager (CMM) acts as a message distributor for the character module (CM). It controls the CM finite-state machine as described in Section 4, and calls the NCA, the CF or the MSA-CIP, depending on whether no recent input was received, information arrives from the speech recogniser or gesture recogniser that input is being received, or a frame arrives from the input fusion module, respectively. The MSA returns a response list to the CMM via the MSAM. The CMM forwards this response list to the response generator. For interpretation of the abbreviations mentioned, the reader is referred to Figure 3.1.

5.3 Conversational intention planner

The conversational intention planner (CIP) receives (i.e. has access to) the entire input frame as delivered by the input fusion module to the character module. The CIP's goal is to plan HCA's next conversational move. The CIP draws on a number of functions to decide on the next move. The main part of these functions may be regarded as agenda points which jointly help determine what HCA will do next. D1.2a describes HCA´s conversational agenda as modelled by the CIP. In the following, we provide a systematic view of the CIP functions. Several of the agenda points or functions are context-dependent and draw on the conversation history. The ordered set of CIP functions are:

- *meta function:* this function checks whether or not to register meta communication. In the former case, it decides which action must be taken and returns the action to the core of the CIP planner. If meta-communication is registered, the meta function also decides whether this should result in an increment to HCA's emotional state (see NICE Report D1.2a);
- *domains function:* this function decides which domain(s) should be addressed next. Based on the action output from the meta function, the domains function checks if meta-communication must be handled and, in this case, which action to take, and checks if any other domain should be addressed;
- *topics function:* this function decides which topic(s) to address next. The decision must be in accordance with the chosen domain(s). The function also checks for emotional increments, such as when. the user has addressed a topic which HCA (dis-) likes;
- *patterns function:* this function checks if there are particular patterns in the input, e.g. in terms of repeated meta-communication or repeated attempts to enter the fairy tale world despite HCA's attempts to talk about other domains. Different patterns will influence HCA's emotional state in different ways. In NICE prototype 2 we also expect that patterns may influence the chosen speech act for the next output if the user continues to be passive and not take initiative;
- *core of the CIP planner:* this is the main part of the CIP, i.e. the part which controls the functions described above. The CIP core calls these four functions in the order in which they are listed above. The CIP core then calls the EC (emotion calculator) with the emotional increments from the functions and gets an updated emotional state in return. Emotion computation is described in more detail in D1.2a. The CIP core then establishes a priority value set on the basis of the different priority variables from the above functions. A priority value set is a list which may contain one or more domain, topic, index, speech act values, i.e. [(domain, topic, index, speech act)*]. The index is an integer which is functionally equivalent to input semantics, i.e. it is used to retrieve a particular output from the knowledge base. The conversation history is updated with the priority value set. Finally, the mind state agent manager (MSAM) is called. The important parameters for the MSAM are emotional state, priority value set, meta pattern length, action, skip_replies, and input domain(s), topic(s), semantics, and speech act(s), cf. Tables 5.2 and 5.1.

5.4 Mind state agent manager

The mind state agent manager (MSAM) is a management module which, on the basis of the CIP results, communicates with one or more domain agents (DAs) to retrieve output. The MSAM first checks the skip_replies variable (Table 5.2). If skip_replies is true, it means that the user input part (domain, topic, etc.) should be ignored and only the priority set value should be processed. If skip_replies is false, the MSAM must first look for a reply (or several replies) to the user's input and then find a continuation based on the priority value set.

Replies to user input are handled as follows. The domain(s) parameter, cf. Table 5.1, includes one or more domains identified in the user's input. If there are several domains, the topic(s), semantics, and speech act parameters, and also the gesture object and gesture type parameters (Tale 5.1), must include a number of topic sets, semantics sets, etc., corresponding to the number of domains, so that it is clear what belongs together. For each domain, the MSAM must identify the corresponding topic(s), semantics, etc., and call the responsible domain agent (DA). The parameters needed by the DA are obtainable from the frame. An example of what is needed is:

- topic (study), semantics what (object), speech act (null), emotional state (0,0,0), action (null), gesture object (Jenny Lind), gesture type (pointing).

Only when the Physical Presence DA or the Meta DA is called, the gesture object and gesture type may be of relevance. If the Meta DA is called, meta pattern length is also relevant. It tells how many times in succession there has been meta-communication.

For continuations, the available information is a bit different from what we have via user input. We never have semantics. Instead, the CIP may have decided on an index number which is functionally equivalent to semantics.

In some cases, the character's emotional state will change based on the retrieved *output.* This can only happen in mini-dialogues, cf. NICE Report D1.2a. Therefore, when a DA retrieves output from a mini-dialogue, it first checks if the output will affect the emotional state. If this is the case, the DA retrieves the emotional increment and sends it to the emotion calculation module which calculates an updated emotional state and returns it to the DA via the frame. The DA then retrieves output from the mini-dialogue, using the updated emotional state. The MSAM will update the history with the emotional state which may have been changed by one or more DAs, and it updates the history with the frame.

After communication with the knowledge base, a DA will return an output reference set, i.e. a set of references for the response generator (RG) to use. The MSAM will store each returned reference set in its response list. When all needed DAs (one or several per turn) have reacted and sent a reference set to the MSAM which has stored these in the response list, the MSAM sends in the frame the response list containing all reference sets for the HCA output of this turn to the CMM which sends it to the RG. A reference set includes information about oral as well as gestural output.

5.5 Conversation history

The conversation history (CHi) keeps track of the context of the conversation in terms of what has happened in previous user-HCA exchanges. Any module of the mind state agent has read/write access to the conversation history. Thus, the conversation history keeps a record of the frame resulting from each exchange, i.e. a frame which includes the user's input, references to HCA's output, and the intermediate values produced by the mind state agent module, cf. above. In addition, the conversation history keeps track of patterns, such as repeated occurrences of meta-communication; it keeps track of whether a domain has been sufficiently covered or still has any conversational issues left; and it keeps track of which topics HCA has not yet talked about.

Additional information is expected to be added to the conversation history during development of NICE PT2.

6 References

6.1 Publications

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: Designing Interactive Speech Systems. From First Ideas to User Testing. Springer Verlag 1998.

Bernsen, N. O.: When H. C. Andersen is not talking back In Rist, T., Aylet, R., Ballin, D. and Rickel, J. (Eds.): *Proceedings of the Fourth International Working Conference on Intelligent Virtual Agents (IVA'2003),* Kloster Irsee, Germany, 2003. Berlin: Springer Verlag 2003, 27-30.

6.2 NICE reports

D1.1: Bernsen, N. O. (Ed.), Boye, J., Kiilerich, S. and Lindahl, U.: Requirements and design specification for domain information, personality information and dialogue behaviour for the first NICE prototype. HLT project NICE Report D1.1. NISLab, Denmark, 2002.

D1.2a: Bernsen, N. O., Dybkjær, L. and Kolodnytsky, M.: Analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen in the first prototype HLT project NICE Report D1.2a. NISLab, Denmark, October 2003.

D3.3: Martin, J.-C. (Ed.), Bernsen, N.O., Blasig, R., Boye, J., Buisine, S., Dybkjær, L., Fredriksson, M., Généreux, M., Gustafson, J., Lindahl, U., and Mehta, M.: Analysis and specification of cooperation between input modalities and cooperation between output modalities. HLT project NICE Report D3.3. LIMSI, France, 2003.

D3.6: Martin, J.-C., et al.: Multimodal input understanding module for the first prototype. HLT project NICE Report D3.6. LIMSI, France, October 2003.

D3.7: Corradini, A., Gustafson, J., Martin, J.-C., Mehta, M., Bernsen, N.O. and Fredriksson, M.: Multimodal output generation module for the first prototype. HLT project NICE Report D3.7. LIMSI, France, October 2003.

D4.2: Fredriksson, M., Königsmann, J. and Johannesson, L.: First version of the system prototype, inhabited by characters from H. C. Andersen's world. HLT project NICE Report D4.2. Liquid Media, Sweden, October 2003.

D5.1b: Boye, J., Gustafson, J. and Wirén, M.: First prototype version of dialogue management and response planning for the fairy-tale domain. HLT project NICE Report D5.1b. TeliaSonera, Sweden, October 2003.