NICE project (IST-2001-35293)

Natural Interactive Communication for Edutainment

NICE Deliverable D1.2-2a

Analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen in the second prototype

29 October 2004

Authors

Niels Ole Bernsen with contributions from Marcela Charfuelán, Laila Dybkjær, Abhishek Kaushik, Mykola Kolodnytsky and Manish Mehta

NISLab, Odense, Denmark

Project ref. no.	IST-2001-35293
Project acronym	NICE
Deliverable status	Restricted
Contractual date of delivery	1 November 2004
Actual date of delivery	29 October 2004
Deliverable number	D1.2-2a
Deliverable title	Analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen in the second prototype
Nature	Report
Status & version	Final
Number of pages	29
WP contributing to the deliverable	WP1
WP / Task responsible	NISLab
Editor	Niels Ole Bernsen
Author(s)	Niels Ole Bernsen with contributions from Marcela Charfuelán (Sections 5.3, 7.1, 7.3), Laila Dybkjær (Section 5.2), Abhishek Kaushik (Section 6.2), Mykola Kolodnytsky (Section 6.1), and Manish Mehta (Sections 6.2, 7.1)
EC Project Officer	Mats Ljungqvist
Keywords	Life-like animated conversational agents, domain modelling, personality modelling, conversation behaviour
Abstract (for dissemination)	This report, Deliverable 1.2-2a from the HLT project Natural Interactive Communication for Edutainment (NICE), describes the analysis and representation of domain information, personality information and conversation behaviour for life-like animated conversational agent H. C. Andersen in the second NICE HCA prototype.

Table of Contents

1 Introduction

1.1 On this report

This NICE report D1.2-2a "Analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen in the second prototype" builds on Report D1.1-2a "Requirements and design specification for domain information, personality information and dialogue behaviour for the second NICE HCA prototype". Both reports are based on work on WP1 "Domain representation and domain-oriented conversational human-machine dialogue", and both reports serve to document the basis for the second NICE prototype (PT2). The reported work on WP1 provides input to WP4 (animated characters) and WP5 (conversational abilities). In particular, the reader is referred to D5.2a "Second prototype version of conversation management and response planning for H.C. Andersen" which builds on the present report, describes the character module components which are not described in the present report, and provides an information flow description of the HCA character module. In addition, WP1 work draws on information from the analyses of data collected in WP2 on interaction between users and incremental versions of the first prototype, cf. Report D2.2a "NISLab's Collection and Analysis of Multimodal Speech and Gesture Data in an Edutainment Application".

Given the differences in domain information, personality, and dialogue behaviour of Hans Christian Andersen (HCA) in his study, on the one hand, and the fairytale characters in their fairytale world on the other, clarity of exposition suggests to SPLIT D1.2 into two parts, one on HCA in the present report (D1.2-2a) and one on the fairy tale characters (D1.2-2b).

1.2 Rationale for the PT2 requirements specification

Quoting from the predecessor (D1.2-1a) to the present report:

> *The PT1 user testing data will no doubt provide important information on the extent to which we have been right or wrong in our common ground assumptions, story-telling strategy, rhapsodic strategies, as well as in the compromise struck between the entertainment and the educational sub-goal of the system.*

As regards the main topics of the present report, the purpose of the PT2 requirements specification section, revisited in Section 1.3 below, was to address key *critical* observations made in the HCA PT1 user test. The purpose, of course, was not to address and try to fix what was not broken, such as, in particular, the high-level theory of conversation underlying HCA's conversation, including its ideas on: unconstrained mixed-initiative dialogue, the demise of a single hard-coded dialogue structure in favour of dynamical output planning, pattern-spotting in the conversation history, a conversational agenda for HCA, the importance of story-telling, common ground, the conversation domains preferred by the users, educational value, entertainment value, and user gesture input. This theory was largely validated in the user test and still forms the basis for PT2.

However, the key critical observations made by the user test subjects were that the users found that

- HCA did not always listen to what they said;
- made too many unwanted repetitions; and
- produced too many unwanted responses.

The cash value of these user criticisms has been analysed in Report D2.2a "NISLab's Collection and Analysis of Multimodal Speech and Gesture Data in an Edutainment Application" and discussed in several publications, e.g., [Bernsen and Dybkjær 2004].

Briefly, following our own analysis of the user test and the PT1 system, and diagnostically expressed in PT1 system terms, what the users correctly perceived was the following issues:

- bugs in conversation management, in particular loops;
- inflexibility of the PT1 mini-dialogue representation and processing approach which too often generates irrelevant output;
- too loose control of conversational continuations, generating irrelevant output;
- lack of appropriate response to generic input, including meta-communication, cf. Chapter 4 below;
- superfluous HCA repeat meta-communication requests when he should have been able to understand the user; and
- HCA's lack of ability to factor in the logical implications of user input in terms of which conversational topics to avoid.

In addition, our own analysis of PT1, of the Wizard of Oz simulations preceding the user test, as well as our incremental development plan for the NICE HCA system, suggest the need for strongly increased emphasis in PT2 on many different, additional aspects of conversation coherence. The user test condition II, i.e., the one in which the users had an agenda to accomplish during conversation, demonstrated the importance, as we had hypothesised, of designing for drive symmetry, i.e., for enabling both HCA and the user to drive the conversation forward as they see fit, avoiding the less important, opposing challenges of building HCA-driven conversation and building a user-driven question-answering machine.

In conclusion, the PT2 requirements specification items relevant to the present report represent the joint impact of (i) our original development-and-test plan, according to which we could not build all our ideas into the first HCA system prototype but had to adopt an incremental approach, (ii) analysis of the many hours of system simulation data gathered in the project so far, and (iii) the PT1 user test including the results of our subsequent analyses of the user test.

1.3 Re-visiting the PT2 requirements specification

It is useful by way of introduction to re-visit D1.1-2a in order to briefly highlight any differences between the specification presented in D1.1-2a and the more specific implementation strategies presented in this report. For this purpose, we reproduce below the relevant parts of the Character module specifications, i.e. Points 20 through 32 from Section 4.5 in D1.1-2a. To these, we have added two new points, i.e. 33 and 34, on PT2 innovations which were not included in the PT2 requirements specification. Requirements met by PT2 are marked by OK. Comments from D1.1-2a have been retained. New comments are marked as *New comments.* Requirements not met by PT2 have new comments attached.

20. Domain and topic-specific ontologies for knowledge and domain representation. **OK**

PT1 uses knowledge-based ontologies already. However, we need a more articulate ontology structuring for HCA's domains and topics in order to support flexible conversation management.

21. Ontology-based capability for evaluating user claims. **OK**

This is one of the advantages of using articulate ontologies, cf. (20).

22. Conversation intention planning based on highly articulate sectorial hierarchies organised as a relevance network. **OK**

This is a crucial set of mechanisms for ensuring conversational coherence: relevance, logicality, absence of obscurity, etc.

23. More articulate, hierarchically organised script and plan rules than in PT1. **OK**

This is another crucial set of mechanisms for ensuring conversational coherence: domain and topic shifts when appropriate, first-things-first when HCA enters a new domain, avoidance of undue repetition, etc.

24. Any-time-anywhere break-in and break-out of mini-dialogues. **OK**

The user can change domain and topic at any time and at any point in the conversation. *New comment: and* the user can break into mini-dialogues at all meaningful junctions.

25. More extensive meta-communication handling compared to PT1. **OK**

This is a main PT2 challenge as well. We need to see how far we can get in PT2 as regards the handling of semantic and referential ambiguity, user clarification handling, system clarification handling, out-of-domain input handling, obscure input handling, etc.

26. Smart handling of back-channelling. **OK**

The decision to de-prioritise barge-in is expected to reduce, but not entirely remove, the problem of handling back-channelling. We need to find generic ways to address the remaining issues.

27. Extended domain/topic coverage, including: more fairytales, something on HCA's other works, more on HCA's person and physical presence, something on HCA's teenage and adult life, more on games, new topic on modern inventions. **OK**.

New comment: however, we have not added any more fairytales because we have prioritised conversation management innovation over developing more-of-the-same. The new topic of modern inventions has been experimentally addressed in a new way together with the topic of HCA's friends and acquaintances, see new Point 34 below.

28. *Desirable:* more use of conversation history patterns compared to PT1.

New comment: this is currently being specified.

29. *Desirable:* machine learning of games and/or modern inventions ontologies.

New comment: this has not been done.

30. *Desirable:* extended user modelling and use of models compared to PT1. **OK**

31. *Desirable:* sense user interest in a domain/topic.

New comment: this has not been done.

32. *Desirable:* praise users when deserved.

New comment: pending for the moment.

New comment: the following innovations are new compared to the PT2 requirements specifications.

33. Typed user interface.

New comment: a typed user interface is currently being added to PT2. This interface serves several opportunities, including over-the-Internet typed conversation with HCA, HCA access by people with speech disabilities, and switching from spoken to typed interaction when required.

34. Augmenting HCA's knowledge with WWW-based knowledge.

New comment: a new module has been specified, built, and tested. The module takes spoken or written system input, uses three top-quality web-question/answering (Q&A) systems to look up the requested information on the WWW, and filters the returns, using only top-quality returns for system output. In this way, users who ask, or otherwise mention to, HCA, names

of famous friends or acquaintances of HCA, or modern inventions, make HCA retrieve knowledge from the WWW on those items. This experimental development is motivated by several ideas, such as the desirability to be able to add knowledge to HCA by simply adding the relevant words to the speech recogniser's vocabulary, enable typed conversation with HCA about topics not in the recogniser vocabulary, and enabling typed interaction with HCA on topics which are not even in the lexicon of the natural language understanding module. In a second step, we plan to integrate HCA's handling of the Q&A returns in his conversation management structure. Due to time constraints, we may not be able to demonstrate the latter in the lifetime of the NICE project.

1.4 Conclusions

In conclusion, with the exception of the minor point that no more fairytales have been added to HCA's knowledge, PT2 meets all the mandatory requirements specified in the PT2 requirements specification document D1.1-2a as well as some of the non-mandatory requirements specified. One requirement, i.e., the one on knowledge of new inventions, is being addressed in a way not anticipated in D1.1-2a. Two new requirements which were not present in D1.1-2a have been analysed, design-specified, and are being implemented at the time of writing, i.e., a typed user interface and a web-based Q&A filter module.

1.5 Underlying innovations

Underlying the fact that PT2 meets its requirement specifications are some major architecture-level innovations in PT2 as compared with PT1. At overall system architecture level, cf. Figure 1.1, two new modules are being integrated, i.e., the Text interface and the Web agent. In addition, speech recognition has been integrated. The innovations also affect the architecture of the natural language processing module which, conforming to PT2 specifications, has been modified in order to produce HCA domain ontology-conformant concepts as described in Report D3.5-2a. More to the point of the present report, the architecture of the NICE HCA PT2 character module has been revised compared to that of the NICE HCA PT1 character module. Figures 1.2 and 1.3 present the PT1 and PT2 character module architectures, respectively. The main difference shown at the level of detail of these two figures is the absence of the Mini-dialogue processor. At a more detailed level, however, the differences are more pronounced. The PT2 knowledge base has a far simpler structure than in PT1. However, the Conversation intention planner, in particular, has a completely new internal structure which enables it to comply with the PT2 Character module requirements, cf. Figure 1.4, D5.2a and below.

In order to satisfy the PT2 requirements as regards analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen, the PT2 character module architecture and knowledge representation embody major modifications. These are:

- the Conversation mover (Cmover) is a new module in the Conversation intention planner, which identifies output(s) corresponding to the concepts received from Input Fusion and rejects outputs which do not correspond to the concepts received from input fusion;
- the Conversation intention planner includes a set of hierarchical and internally linked domain ontologies which all follow the same overall design principles, i.e., what is called a (within-domain and cross-domain) *relevance network* in the PT2 specification. This enables the Conversation intention planner to produce relevant conversation continuations at any time;

- the finite-state machine-processed mini-dialogues in PT1 have been replaced in PT2 by mini-dialogue structures with anytime break-in and break-out properties. This ensures that HCA is capable of following the user's input wherever it may lead in the discourse context. It should be noted that the general mini-dialogue philosophy is still present in PT2, i.e., the idea of allowing HCA, on occasions at which he takes particular interest in the user's input, to carry our in-depth conversation on certain topics. It is the anytime break-in and break-out functionality which has been achieved in PT2;
- the reasoning capabilities of the PT2 Cmover Post-processor helps the Conversation intention planner select the right output in context and adds to the reasoning power of the domain agents;

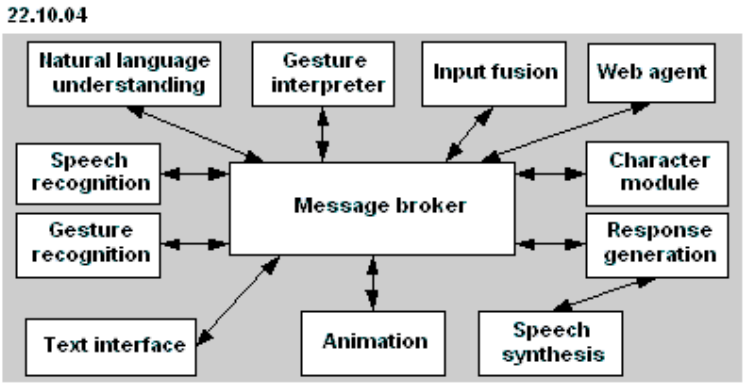

Figure 1.1. Overall NICE HCA PT2 system architecture.

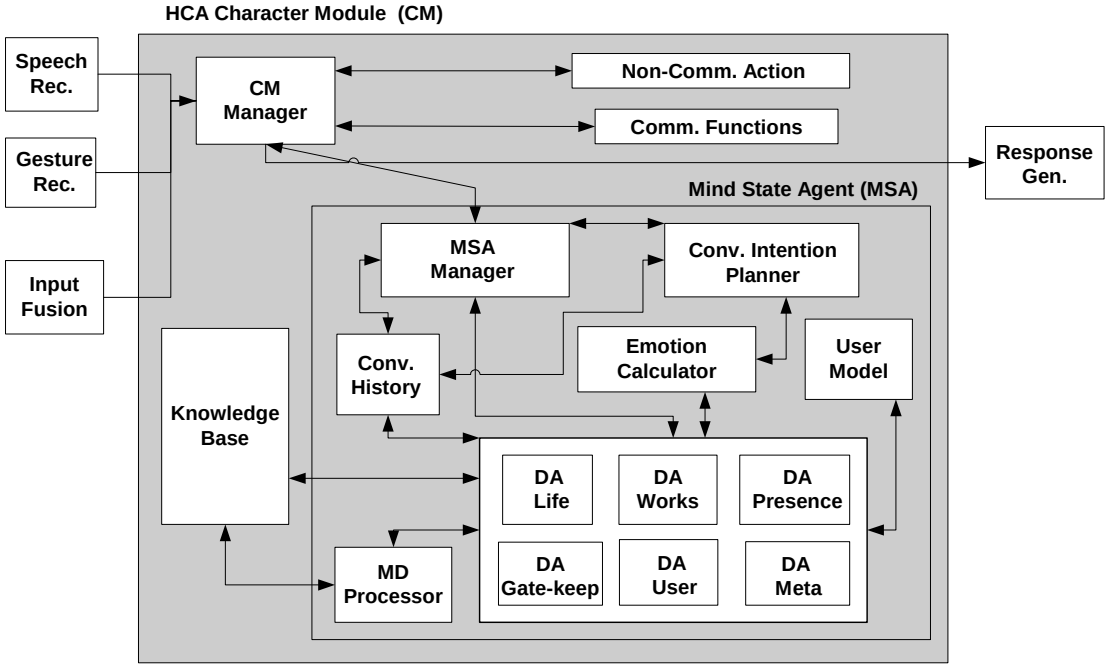

Figure 1.2. HCA character module architecture for the first NICE HCA prototype.

- in the PT2 Conversation intention planner, HCA now has less strong conversation plans overall than in PT1, enabling the user to drive forward the conversation much harder than was the case in PT1. Thus, HCA has less opportunity than in PT1 to monopolise the current conversation in order to stick to his favoured conversation topic at the moment; and
- the PT1 relational database has been replaced by a comparatively simpler output database, significantly shifting the burden of hierarchical and linked ontological domain representation from the knowledge base to the Conversation intention planner's new Move processor and Knowledge structure modules.

Jointly, and together with the new concept-based approach to natural language understanding, the above innovations enable the Character module to satisfy the key PT2 requirements 20 through 26 in Section 1.2 above.

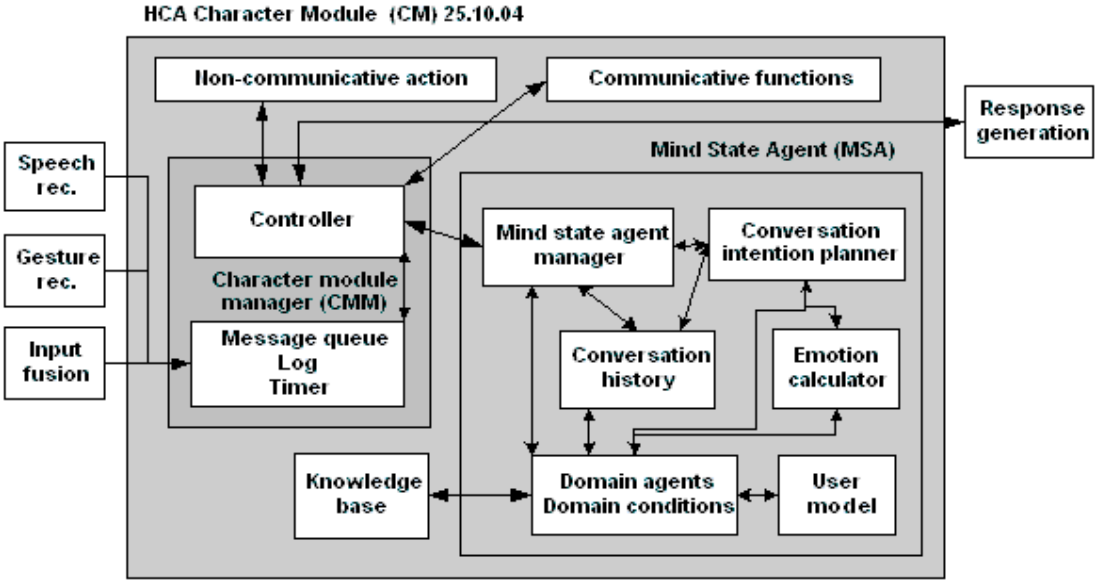

Figure 1.3. HCA character module architecture for the second NICE HCA prototype.

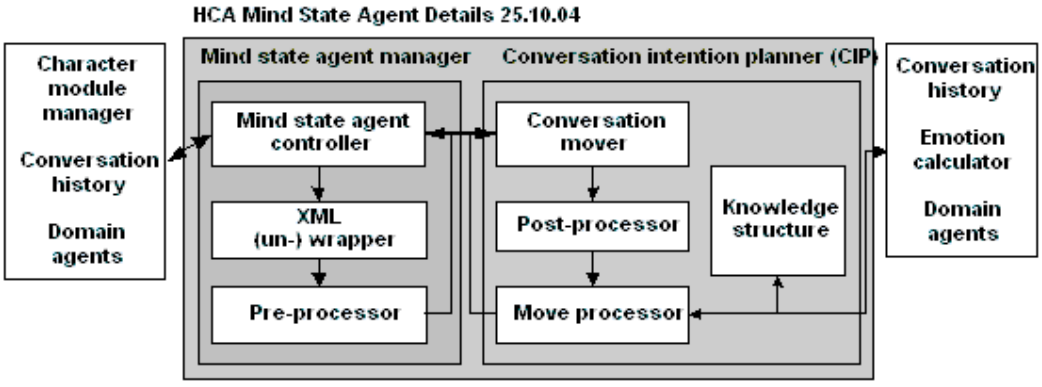

Figure 1.4. HCA PT2 Mind state agent details.

The following sections describe in more detail the PT2 solutions to the requirement specifications in Section 1.2. The section headings refer to the relevant requirement specification numbers in brackets.

2 Domain-specific ontologies for knowledge and domain representation (20), ontology-based capability for evaluating user claims (21), more articulate, hierarchically organised script and plan rules than in PT1 (23), any-time-anywhere break-in and break-out of mini-dialogues (24), smart handling of back-channelling (26)

In PT2, HCA's domain knowledge is represented as dynamically linked domain ontologies. Each domain is represented as a hierarchically organised, dynamically linked ontology. The leaves of the hierarchies are output references which can be retrieved from the Knowledge base and sent to the Response Generator.

The PT2 domains are: HCA's works, with emphasis on his fairytales, his life, with emphasis on his childhood, his physical presence and personality, his study, the games he likes or is interested in, the user, his role as gatekeeper for the fairytale world, and meta-communication. Figure 2.1 shows a fragment of the Life ontology hierarchy.

- life domain introduction story1 + continuation_HCA/user, life domain introduction story2_HCA/user + continuation HCA/user, life domain introduction story3_HCA
    - lifetime sub-subsegment story
    - birth date sub-subsegment story
    - death date sub-subsegment story
  - family segment story + Life continuation pause + continuation
    - father subsegment story + Life continuation pause + continuation
    - mother subsegment story + Life continuation pause + continuation
    - grandfather subsegment story + Life continuation pause + continuation
    - grandmother subsegment story + Life continuation pause + continuation
  - childhood segment story + Life continuation pause + continuation
    - where I lived subsegment story + Life continuation pause + continuation
    - how we lived subsegment story + Life continuation pause + continuation
    - my school years subsegment story + Life continuation pause + continuation
    - my childhood games subsegment story + Life continuation pause + special continuation + mini-dialogue + continuation

Figure 2.1. Fragment of the PT2 Life ontology hierarchy.

All PT2 domains have a format resembling the one shown in Figure 2.1 but with some individual differences required by each domain, such as the size and depth of the hierarchy, or the number of embedded mini-dialogues. This format of domain representation has a number of important properties which serve to meet the requirements to PT2 conversation, including:

- hierarchical organisation into segments providing increasingly detailed information from the top down;
- easy design-time addition/deletion of segments or segment parts, including mini-dialogues, without modifying the general domain processing algorithm;
- HCA can enter conversation about a domain several times, using new introduction stories each time;
- all outputs used are blocked so that HCA will not use them any more unless the user makes HCA use them;
- the user can enter conversation about any part of any domain at any time;
- some parts of the domain ontology can be entered by the user-only, such as 1.1.1 through 1.1.3 in Figure 2.1. For these segments, no continuation will be used;
- the domain segments which can be entered by HCA are all dynamically linked by continuations, the continuations being specified through variables, thus taking care of blockings already made: the variables enable the Conversation intention planner to select the next segment, either at the existing hierarchy level or higher, which has not been blocked. This makes the domain ontology resemble, to some extent, a sub-domain mini-dialogue, cf. below;
- each story told by HCA is followed by a pause;
- each pause is specified such as to enable certain user inputs, such as back-channelling, to take place without disrupting the system's dynamical selection of a relevant continuation;
- each pause is specified such as to enable domain or topic changing user input to actually change domain or topic;
- the domain ontology may include not only output accessible to the user-only but also embedded mini-dialogues;
- since the mini-dialogues include pauses as well, the user may leave a mini-dialogue at any time;
- since the mini-dialogues include stories as well as more contextual mini-dialogue conversation exchange operators, the user may enter a mini-dialogue at any point at which HCA tells a substantial story;
- in general, domains are processed using closely similar algorithms;
- each of HCA's domain representations (ontologies) constitutes a domain design (or pattern) library volume which, we hypothesise, can be easily re-used for other characters by replacing the HCA output with output for the new character while retaining the domain processing algorithm;
- due to the design of the conversation mover, any relevant user claim is evaluated by consulting the appropriate ontology.

3 Conversation intention planning based on highly articulate sectorial hierarchies organised as a relevance network (22)

The sectorial hierarchies are the domains described in Section 2. In addition to managing conversation at domain level, the Conversation intention planner manages the conversation at the higher, supra-domain level. This latter task has at least two main aspects, i.e. managing HCA's conversational agenda and managing domain change at the initiative of either the user or HCA.

3.1 HCA's conversational agenda

Basically, the conversational agenda goals are to:

- respond to the user's input, and
- decide on a continuation of the conversation.

A *continuation* is output which serves to continue the conversation rather than simply responding to the user's input and thereby leaving all or most initiative with the user. A continuation may be either a question or a statement.

Overall, in PT2, HCA's conversational agenda is largely similar to the one in PT1. The main differences are that, in PT2, HCA no longer insists to talk at length about his fairytales and that his agenda management has more subtlety than in PT1. Thus:

- reflecting human conversational strategy, HCA assigns top priority to resolving miscommunication problems through meta-communication before proceeding with the conversation. When miscommunication is detected, e.g., due to low recogniser confidence score or need for clarification, the Meta domain takes priority over all other domains. HCA will not try to respond to the subject brought up by the user but continue the conversation through the meta-communication output;
- HCA wants to talk about the User domain early on in the conversation in order to gather information about the present user before proceeding with the conversation. If the user breaks out of the User domain, HCA will respond to what the user says and then return to the User domain;
- contrary to the case of PT1, HCA has no further domain preferences in PT2. Rather, if HCA is to choose a new domain of conversation, he will dynamically choose the, or a, domain which has been addressed less than other domains so far during conversation;
- HCA wants to have talked about all of the domains of User, Life, Works, HCA and Study before addressing the Gatekeeper domain. The Gatekeeper domain concerns how the user can take leave of HCA in his study and enter the fairytale world. Therefore, if the Gatekeeper domain is introduced by the user before the other domains have been sufficiently covered in the conversation, HCA will try to escape that domain by proposing a topic from one of the other domains.

3.2 Managing domain change

This section describes the notion of a relevance network, i.e. the representation of HCA's complete knowledge as a set of dynamically linked domain hierarchies. This set is only present in the Conversation intention planner and no longer, as in PT1, in the Knowledge base as well. Rather, the set includes references to the PT2 Knowledge Base from which the

Conversation intention planner retrieves output when it has decided which output to retrieve. The output consists of a response to the user's input and, in many cases, a dynamically selected continuation output which HCA uses to move the conversation forward. Each output is a reference to combined spoken and non-verbal output to be generated by the HCA response generator.

If the user changes domain, this is detected by the Conversation mover. The Conversation intention planner finds the new domain addressed, including the appropriate output. If HCA decides to change domain, the Conversation intention planner uses a prioritising algorithm to identify the domain to address next and uses the existing blockings already made in the chosen domain to identify his new conversational contribution.

To ensure conversational variety and liveliness, HCA will only sometimes use a continuation. Thus,

- if a mini-dialogue is ongoing, the next output specified in the mini-dialogue structure may be viewed as the continuation. No other separate continuation is used. If the user changes domain or topic, HCA will always follow the user;
- if meta-communication is needed, the meta-communication output may be seen as a continuation. No other separate continuation is used;
- if the user addresses a reply-only sector in the domain hierarchy, no continuation will be used, cf. Section 2.1.

Finally, a more general strategy has been implemented in order to increase the conversational variety and liveliness of HCA's contributions. For a range of conversational input topics which we expect will be addressed with relatively high frequency in conversations with users, HCA has several different response variations available. Some obvious cases are HCA's opening and farewell greetings, and the mini-dialogue about the User domain, but the strategy has been implemented for other domains as well.

In addition to the cases above there will, of course, be cases in which HCA is not able to respond to the substance of the user's input because he does not have the knowledge required. Although the user's input may have been recognised and parsed correctly, HCA obviously does not have a to-the-point reply for each possible input understood by the system. This follows from the *systematicity* property of the natural language understanding module. For instance, if the linguistic input parser can generate appropriate semantics for "I like your fairy tales" (which it can) and if HCA's 'mother' is in the lexicon, then the parser will also generate appropriate semantics for "I like your mother". However, HCA does not have any to-the-point response to the latter input utterance.

If HCA cannot respond directly and relevantly to the substance of the user's input, it is important that he is able to respond at some more abstract level, so that the conversation does not just stop. We are making substantial efforts to address this problem, cf. Section 4 below.

4 More extensive meta-communication handling compared to PT1 (25), more use of conversation history patterns compared to PT1 (28), praise users when deserved (32)

Meta-communication is communication about the communication itself and is normally made in order to overcome miscommunication. Meta-communication can occur at any point during conversation. In this regard, meta-communication may be called *generic communication.* The notion of generic communication is, however, much wider than that of meta-communication, because there are many other exchanges which may occur at any point during conversation. It remains a research challenge to identify all possible generic communication for the purpose of building high-quality conversational systems.

In PT2, substantial efforts are being made on the important issue of generic communication. The new ontology-based Conversation mover and the new Conversation intention planner's improved grasp of the conversation (discourse) context are crucial to the success of these efforts. To mention just a single example, the system should be able to appropriately handle in context at least three different kinds of user "yes" or user confirmation input, whatever its surface language expression:

- basic yes to a system yes/no question
- basic backchannelling yes
- non-basic yes of both kinds, i.e. user confirmation which includes, in addition, contextual (topic, theme) information, such as "I would like to hear that story" or "That's true, I think".

4.1 Meta-communication

The HCA PT1 system had limited meta-communication abilities. The system primarily handled user and HCA repeat requests, no user input, no-object gesture input, as well as some conversation history-based cases of prolonged miscommunication. In the latter case, the conversation history keeps track of repeated miscommunication and the Conversation intention planner ensures that HCA's output is modified to increase the likelihood of recovering from growing crises in conversation.

The PT2 system has capabilities for handling no user input, user and HCA repeat requests, user and HCA correction, and user clarification. In addition, we are currently specifying strategies for handling the large space of cases in which HCA cannot find output matching the user's input, including recogniser out-of-domain (OOD) input, incomplete input, low speech confidence input, obscure input, parsing failure input, input which does not match HCA's knowledge, etc., as well as repeated cases of all of the above.

4.2 Other generic communication

Other generic communication addressed in PT2 includes: opening greeting, end greeting, yes, no, don't know, user and HCA praise, other user reactions, such as "That's scary", user thanks, and user insult.

5 Extended domain/topic coverage (27), extended user modelling and use of models compared to PT1 (30)

5.1 Domain coverage

Compared to PT1, PT2 HCA knows more details of the three fairytales The Princess and the Pea, The Ugly Duckling, and The Little Mermaid. His knowledge about his life has been split into a number of shorter stories and he now also knows a bit about his youth and adult life. He has substantially more knowledge about himself and his study. Se also Section 6.

5.2 User modelling

As regards user modelling, PT2 HCA still models and uses, like PT1 does, basic information about the user, i.e. age, gender and nationality. In addition, the user model is being used in PT2 in the fairytales domain to avoid that HCA asks questions of the user that require knowledge about his fairytales if the user does not know any of his fairytales. When the works domain is entered the first time, HCA will ask the user as soon as possible if s/he knows any of his fairytales. The answer (in terms of yes/no) is stored in the user model. This information is consulted before HCA asks certain questions about his fairytales later in the conversation. If the user has told that s/he does not know any of his fairytales, HCA will refrain from asking questions which assume such knowledge.

5.3 The PT2 knowledge base

Compared to the PT1 relational knowledge base, the PT2 knowledge base has a simpler structure. The main reasons are (i1) that the PT2 knowledge base no longer has to support a mini-dialogue processor module by including the mini-dialogue structures, and that (ii) the cumbersome PT1 semantic equivalence class handling and maintenance by the knowledge base has been replaced by the Conversation mover's handling of ontology-based Natural language understanding and gesture interpretation concepts.

The – compared to PT1 - more elaborate PT2 domain ontologies are still represented in the knowledge base as domain conversation structures and mini-dialogue conversation structures, cf. the structure fraction shown in Chapter 2. Like in PT1, the knowledge base is queried by the Domain agents. As regards SQL querying, the PT2 knowledge base exhibits two main differences from the PT1 knowledge base:

- there are now two standard queries-only, compared to the more complex set of queries in PT1; and
- the most common PT2 knowledge base query uses the simple string language of the Conversation mover output rather than two sets of complex ids, one from the PT1 Mini-dialogue processor and one from the PT1 relational database itself.

The two generic queries made to the PT2 knowledge base are:

- initialisation-time query for uploading the domain ontologies into the Conversation intention planner Knowledge structure (see Figure 1.4);
- query to retrieve the next conversational move.

The parameters for the second of the two query types just mentioned are:

- output from the Conversation mover, i.e., a conversation move;
- ontology level of the conversation move, i.e., domain, segment, sub-segment, etc.;

The information retrieved in the second of the two query types above is:

- Response generator id for output (system move) to be generated;
- particular variation of output to be generated, if any;
- emotional increment, if any, associated with the output;
- actions to be done by the Conversation intention planner, if any;
- continuation link, if any, to another conversation move;
- Character module expectations regarding the next user input, if any.

6 Typed user interface (33), augmenting HCA's knowledge with WWW-based knowledge (34)

This section describes two new technologies which have been added to PT2 and which were not present in the PT2 requirements specification. The first technology augments the multimodality of user-system interaction, the second augments the system's knowledge.

6.1 Typed user interface

If the NICE HCA system is viewed as an application of a generic application type, it is easily conceivable that this or other applications could be used for purposes, with certain groups of users, and/or in environments in which it would be useful to, e.g.:

- provide the user with alternative linguistic conversational input modalities, such as spoken input and typed text input;
- provide some user group, such as people with speech disabilities, with the typed text input modality option-only;
- provide all users with the typed text input modality option-only, such as when having conversation with HCA over the WWW;
- provide the user with joint or alternative linguistic conversational output modalities, such as spoken output and typed text output;
- provide some user group, such as young people having English, or any other I/O language, as second language, or the hearing impaired, with the typed text output modality.

In order to prepare for, and experiment with, scenarios of use such as those just described, a Subtitles module has been developed and is currently being integrated into PT2. The architecture of the typed user interface component is shown in Figure 6.1.

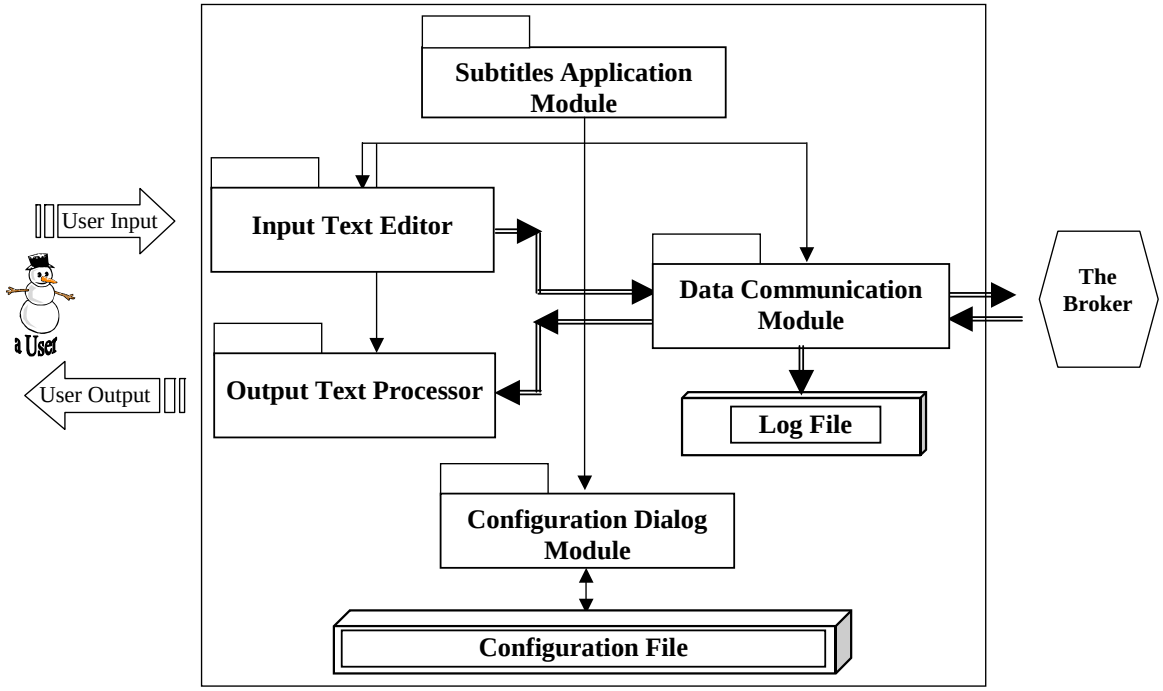

Figure 6.1. Typed interface component architecture.

The main objective of this module is to provide an alternative modality of information input/output for the NICE HCA system. Essentially, the Subtitles module enables the user to type keyboard conversational contributions to the system and enables the system to present the generated linguistic output as text on the screen. The typed input is processed by the system just like the spoken input. The Subtitles module can be customised by the system administrator by using a configuration dialogue window. The system administrator is able to enable/disable and configure the input and output text windows in terms of size, position, font, colour, and text scrolling properties, and to save the configuration for later use. When entering input text, the user is able to use standard text editing operations, such as insert, delete, copy, etc.

6.2 Web agent

One of the bottlenecks for the future development of NICE HCA-style technologies is the provision of sufficient amounts of contents to the system for it to be fun, entertaining, educational, etc. to interact with at length. In our experience during 2.5 years of HCA system development, contents acquisition, preparation, implementation, and testing is just as labour-intensive as we expected from the start of the project. At the start of the project, we anticipated to consider using one or both of the following contents-handling strategies:

- manual contents acquisition, preparation, and implementation;
- contents acquisition through machine learning.

At the present time, we have mainly used the first method above. Machine learning has only been used to a limited extent, i.e. by having the user modelling module learn facts about the user and user preferences. We are still to experiment with machine learning of non-user contents. However, during the project, we have invented a third approach to contents handling, i.e.:

- web-based contents handling.

The WWW includes huge amounts of knowledge, some of which is relevant to the common ground topics shared by HCA and his users. If HCA could tap into this knowledge dynamically and on-line, we would have found a way of avoiding laborious manual contents provision. To explore this approach, we have implemented and integrated into PT2 a Web agent module which uses three top-quality web Q&A (question-answering) systems to search the web for appropriate responses to selected user input. The response candidates retrieved are subsequently filtered by the Web agent module in order to select only top-quality and to-the-point responses to user input. This user input is then sent to response generation.

The input contents which HCA can handle in this way depends on the input modalities used. Thus, (i) if the speech recogniser is being used, the relevant words and phrases must be present in the recogniser's vocabulary and the recogniser's language model must be trained to accommodate the surface language in which the user expresses the contents in question. For the moment, the recogniser has words pertaining to names of HCA's famous friends across Europe, various kinds of games, and old and new inventions.

If (ii) text-only input is being used, cf. Section 6.1, the possibilities of using the web agent increase substantially. It is no longer necessary to include the relevant words and phrases in the recogniser nor to train the language model. Rather, with no loss of robustness in the system's input language processing, only the Natural language module needs to include the relevant words and phrases in its lexicon.

It is even possible (iii) to process typed input which is not known to the Natural language module. When such input arrives, the Natural language module sends it to the Web agent which investigates possible useful responses from the WWW.

So far, the Web agent technology has been only preliminarily tested. If the technology turns out to work sufficiently well, an important next step is that of further integrating the Web agent into the HCA system. The integration goal will be to dynamically integrate any Web agent-retrieved responses into the systems conversational structure in order for HCA to achieve full mastery of the retrieved output in context.

The basic approach used is to pass the question asked by the user on to some online Q&A engines. Based on certain filtering criteria, the best answer among several answers received is returned back to the user. A basic model of the Q&A agent is shown in Figure 6.2.

The module *QA_Systems* is responsible for sending the query to the online QA engines, viz. *AnswerBus*, *START* and *AskJeeves*, and retrieve answers from them. The basic functionality of the QA_Systems module is same for all three QA engines except for the way in which the raw answer received is processed. The basic design of a *QA_System* module is shown in Figure 6.3.

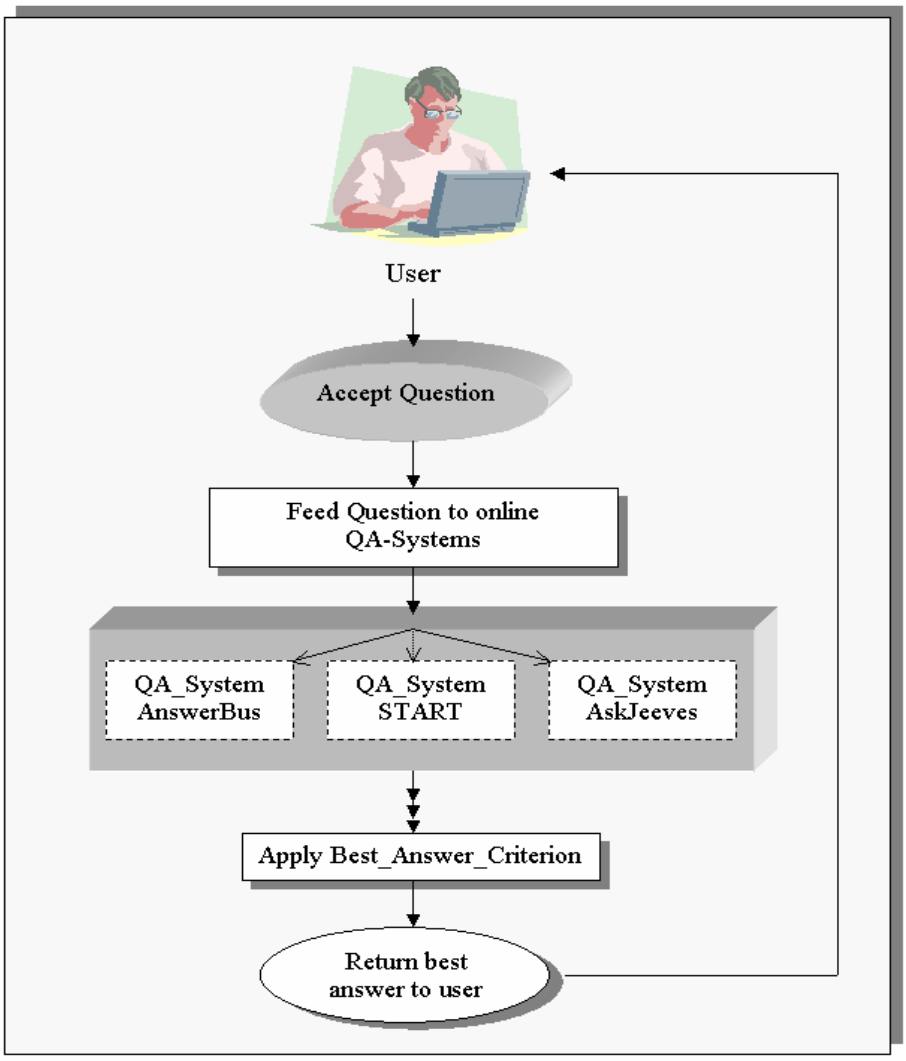

Figure 6.2. Outline of the QuestionAnsweringAgent – QA^2.

The key function of the *Process Response* module is to extract an answer from the raw response received from the engine. The QA Engines obtain their answers by parsing the source file of the relevant HTML pages obtained from the Web for the question in

consideration. Due to certain errors in the HTML source pages and the non-robust nature of HTML parsers, some irrelevant answers are also sometimes retrieved by these engines, which may not have much semantic meaning or which cannot be classified as "spoken English" answers. Though certain Engines like START remove such errors to return natural language answers, others, like AnswerBus, do not perform any kind of error checking. Hence, there is a need to filter out, as much as possible, these answers and retain only the relevant ones.

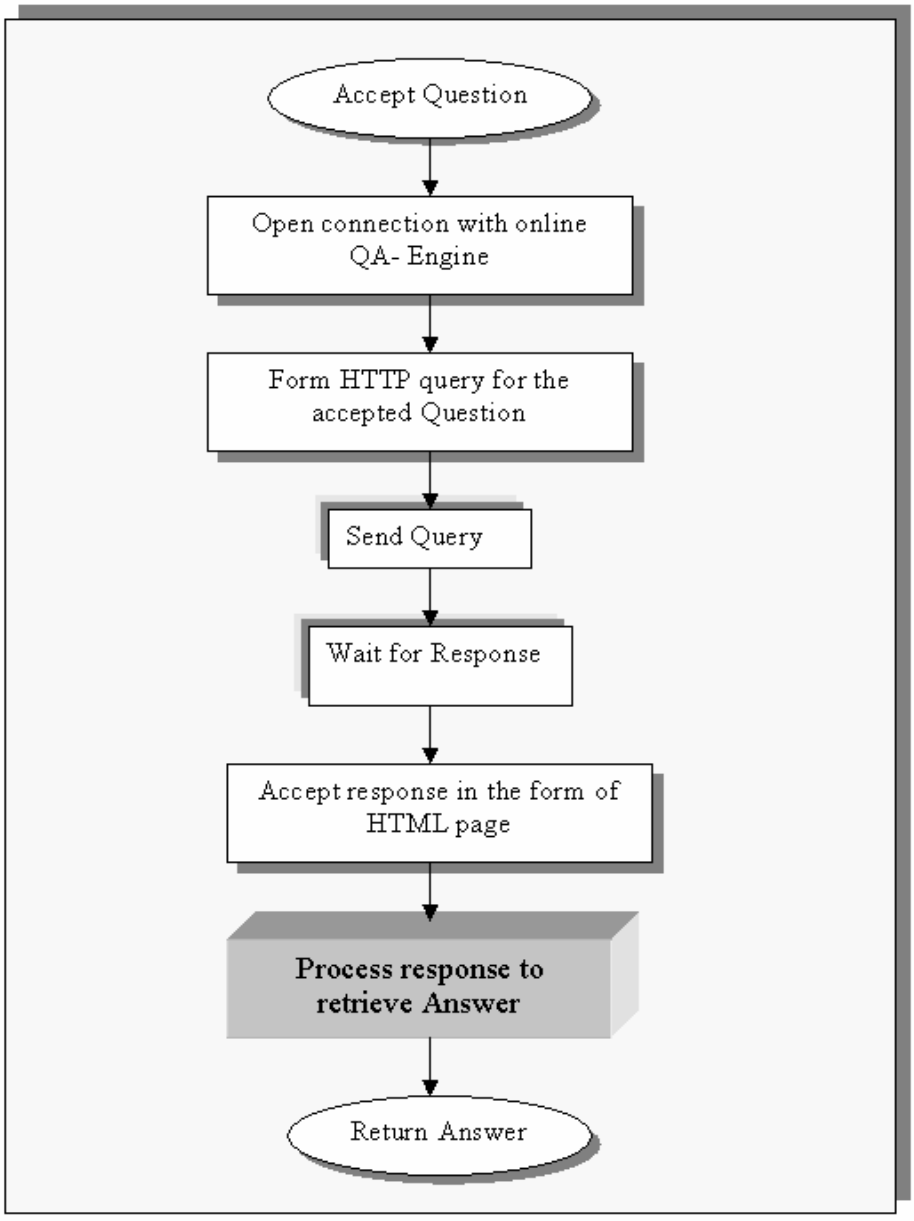

Figure 6.3. Basic design of a QA System.

The filter works by eliminating answers containing certain *stop words*. The stop words are categorised into three categories:

1. stop words which are common words in English or in the WWW dictionary. These words contain only the numerals *0-9* and English alphabet *a-z*;
2. stop strings which are strings consisting of alphabets, numerals and special characters;

3. stop characters which are special characters that should not occur more than a particular number of times in the answer.

An additional strategy is incorporated in the AnswerBus Engine to eliminate irrelevant answers. Keywords from the question are obtained by eliminating commonly occurring English words from it. In case any of the keyword occurs more than α number of times, it is quite probable that the answer is not of much sense. Hence it is neglected. The parameter α can be set based on experimental results. The text file *stopwords.txt* contains the list of these stop words, which can be constantly updated depending upon the user experience with the QA System. A snapshot of the file is shown in Figure 6.4.

```
//Words which should be removed from the Query:

as at and an are also about be because but by do did does don't else from for get got he how has
had have if is it in let my may maybe might not now ok of or on put she such so to the that this
there their these then them than us under were was with why where what who you your yours

//Words which should not occur as complete words in answers:

amp adult align archive breasts breast bgcolor border bin com cgi cd cd's color click cellpadding
cellspacing dot ddd dpd dvd dvd's dvd download erotic earchive email e-mail font ffffff free
gallery galleries gifs gif height http html htm href hurry img image images jpg jpgs jpg's jpeg
jpegs jpeg's linkexchange mpdn mpd meta mp3 mp3's npd nrgm ndpm nude naked netscape nbsp
org page ppd porn pics pic picture pictures quot rgm rhv sex spu srg screensaver site src
screensavers ttt top table vcd vcd's verdana valign wallpaper website wallpapers width www

//Strings which should not occur anywhere in the answer:

www. .com .org .net .asp .htm .html .jsp

//Special characters which should not occur more than ø times in the answers:

1 = + ?
2 - : ; / \ *
3 ( ) { }
4 "
```

Figure 6.4. *stopwords.txt*

The answers received from all the QA engines are passed through the *Best_Answer_Criterion*, which finds the best answer among them. The strategy developed to find the best answer is based on a thorough study of the types of answer that are retrieved for a variety of questions. The basic design of the strategy is shown in Figure 6.5.

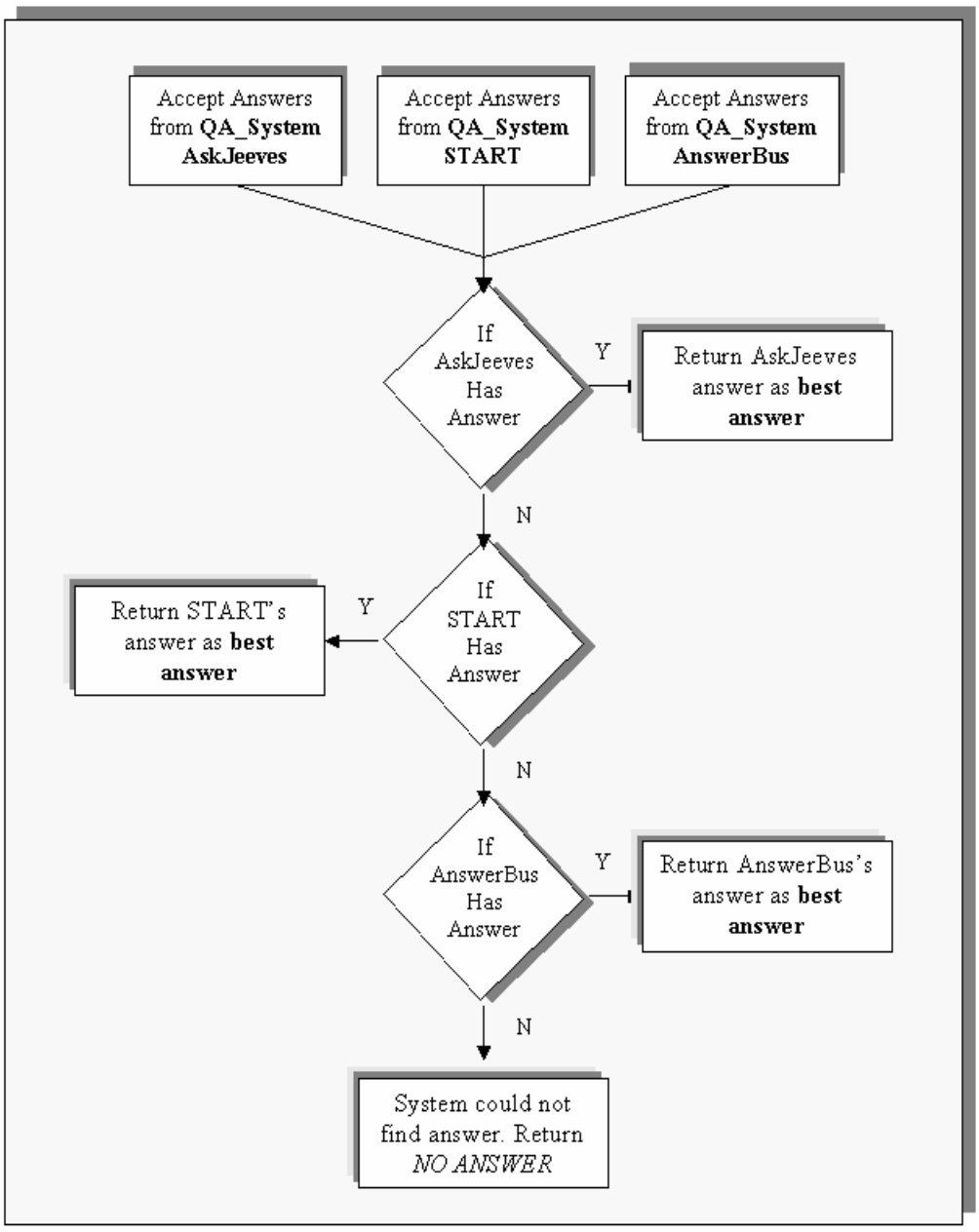

Figure 6.5. Best_Answer_Criterion

7 Other aspects of domain information, personality information and conversation behaviour for PT2 HCA

Chapters 1 through 5 have focused on describing how PT2 satisfies the requirements specification in NICE Report D1.1-2a "Requirements and design specification for domain information, personality information and dialogue behaviour for the second NICE HCA prototype". Chapter 6 described two technical innovations which go beyond the PT2 requirements specification.

Something more needs to be added to the present report, however, because, clearly, its title "Analysis and representation of domain information, personality information and conversation behaviour for H.C. Andersen in the second prototype" comprises more than the issues described so far. The reason that these additional issues were not part of the PT2 requirements specification simply is that the issues have already been basically solved in PT1 and the solutions re-used in PT2. In other words, the PT2 requirements specifications focused on planned PT2 innovations compared to HCA PT1. For the sake of completeness, the relevant system aspects which have been basically re-used in PT2 are briefly reviewed below. "Basic re-use" is compatible, of course, with the fact that all or most of the functionalities in question have been modified and often augmented for PT2 purposes.

7.1 Domain reasoning

In the present context, 'domain reasoning' means reasoning about (i) domain-specific user input, i.e. input which has already been identified by the HCA character module as pertaining to a particular domain among HCA's six knowledge domains, (ii) supra-domain reasoning, or (iii) discourse reasoning, i.e., reasoning about the properties of user-HCA interaction as discourse. The term "reasoning" is less simple to define. Below, we have taken a broad view of reasoning, sometimes, very likely, including functionalities which some might argue are too simple to constitute system reasoning proper and which should rather be termed 'handling' or 'processing'.

In HCA PT2, domain reasoning is being performed by: the Conversation mover Post-processor (Figure 1.4), the Conversation intention planner Move processor (Figure 1.4), and the domain agents (Figure 1.3). The following lists are preliminary ones as the functionalities depend on partner software still to be delivered or on remaining implementation issues wrt. of our own PT2 software, in particular, meta-communication and generic communication issues, since these are necessarily implemented at the final stage.

7.1.1 Conversation mover Post-processor reasoning

The Conversation mover Post-processor (Figure 1.4) has knowledge about how to handle the various types of Conversation mover outputs. The following types of domain and discourse reasoning are being performed by the Conversation mover Post-processor:

- handling of under-determined no input/output matches, i.e., cases in which the Conversation mover does not manage to find a next system move based on the user's input. These cases include:
    - low recogniser confidence input identification;
    - identification of no-match questions other than the over-determined questions below;
    - empty frame identification, if, indeed, this case will exist in PT2;

- handling of multiple input/output matches, i.e., cases in which the Conversation mover finds several outputs matching the input;
- handling of over-determined no input/output matches. At present, this category includes the handling of when-, where-, and why-questions which do not match any key semantics;
- handling of perfect input/output matches, identification of cases of relevance to a particular Domain agent, such as the Meta-communication agent or the User agent.

7.1.2 Conversation intention planner Move processor reasoning

Like the Conversation mover, the Conversation intention planner Move processor(Figure 1.4), has a major role in the Character module. It knows HCA's conversation agenda, the discourse context or the current conversational state, the conversation history, and HCA's emotional state, and it communicates, in addition, with the Domain agents in order to rely on their reasoning and querying of the Knowledge base.

The following types of context-dependent and dynamical domain reasoning are being performed by the Conversation intention planner Move processor:

- conversation agenda reasoning relative to the present input in order to decide how to continue the conversation based on the conversation agenda (dynamic linking);
- discourse context reasoning relative to the present input in order to decide whether and how to react to the input;
- discourse context reasoning relative to the present input in order to decide how to continue the conversation based on the discourse context (dynamic linking);
- conversation history-based reasoning relative to the present input in order to decide how to react to the input;
- reasoning about when and how to update HCA's emotional state based on the current user input and the Conversation history;
- updating of HCA's knowledge structure as a function of HCA's output, including blocking of output used as well as blocking of output implications;
- reasoning about expectations concerning the next user input.

7.1.3 Domain agent reasoning

Basically, the domain agents (Figure 1.3) pre-process knowledge base queries, query the knowledge base, and post-process knowledge base returns, taking into account domain differences. Such differences include differences between the Meta domain and other domains, the handling of gesture input, possibly combined with spoken input, in the Study domain, and the handling of user modelling input in the User domain.

The following types of domain reasoning are being performed by the PT2 domain agents:

- meta-communication reasoning based on the current user input and Conversation history information, cf. Section 4.1;
- other generic communication reasoning, cf. Section 4.2;
- reasoning about gesture objects;
- reasoning about user modelling;
- retrieval of conversational contributions from the knowledge base based on user modelling, emotional state, conversation history, and Conversation intention planner input.

7.2 Analysis and representation of personality information for HCA

HCA's personality as described for PT1 in NICE Report D1.2-1a is primarily represented in the HCA Character module in two ways, i.e. in:

- HCA's conversational agenda represented in the Communication intention planner;
- HCA's dynamic emotional state calculated in the Emotion calculation module.

In addition, the Conversation history stores HCA's part of conversational memory (Figure 1.3) whereas the Conversation intention planner Knowledge structure (Figure 1.4) stores another part of HCA's conversational memory. The PT2 conversational history is described in NICE Report D5.1-2a.

7.3 Emotional behaviour planning

As in PT1, HCA has four *main emotions* in PT2: happiness, friendliness (default), sadness, and anger. Like humans, HCA can be more or less happy or angry, etc., depending on the user's current input, HCA's pre-current input emotional state, and the conversation history. At any time, HCA's *emotional state* is a vector in a space whose *limiting states* are full anger, full sadness, and full happiness. Friendliness, HCA's default emotional state, is located at the "centre" of that space (Figure 7.1). HCA's emotional state becomes modified through emotional *increments* produced by the user's current input and possibly affected by the conversation history as well. When the current user input does not produce an emotional increment, neither by itself nor as a compound effect of the conversation history, HCA's emotional state *decays* towards the default state. These aspects are as described in more detail in NICE Report D1.2-1a.

Factors which may contribute to emotional increments include, apart from emotional decay when no other factors are present, multiple occurrences of meta-communication, multiple attempts by the user to talk about the Gatekeeper domain before the other domains have been sufficiently covered, insulting input from the user, and multiple inputs concerning topics which HCA likes or dislikes.

The following constraints determine the system's use of information on HCA's current emotional state: HCA's emotional state is stored in the Conversation history. The emotional increment pertaining to the current user input, if any, is stored in the Knowledge structure. The conversation history stores any past user inputs which, in combination with the current user input, may affect HCA's emotional state. Finally, HCA's choice of verbal and non-verbal output must reflect his current emotional state as occasioned by the current user input. The chosen processing solution which satisfied these constraints is that the Emotion calculator updates HCA's emotional state based on the current user input and the conversation history, before the Conversation intention planner Move processor selects HCA's output. In this way, output selection can be made to conform with HCA's current emotional state through a single pass through the system. The actual, emotional state-conformant realisation of HCA's output is carried out by the Response generator.

Figure 7.1. HCA's emotional state vector space.

8 References

8.1 Publications

Bernsen, N. O. and Dybkjær, L.: Evaluation of Spoken Multimodal Conversation. *Proceedings of The Sixth International Conference on Multimodal Interaction, ICMI 2004*, Pennsylvania, USA, October 2004. New York: Association for Computing Machinery (ACM) 2004, 38-45.

8.2 NICE reports

D1.1-2a: Bernsen, N. O. et al.: Requirements and design specification for domain information, personality information and dialogue behaviour for the second NICE HCA prototype. HLT project NICE Report. NISLab, Denmark, April 2004.

D1.2-1a: 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. NISLab, Denmark, October 2003.

D2.2a: Bernsen, N. O., Dybkjær, L., and Kiilerich, S.: NISLab's Collection and Analysis of Multimodal Speech and Gesture Data in an Edutainment Application. HLT project NICE Report. NISLab, Denmark, April 2004.

D3.5-2a: Mehta, M. et al.: English natural language understanding for the HCA PT2 system. HLT project NICE Report D3.5-2a. NISLab, Denmark, November 2004 (to appear).

D5.2a: Bernsen, N. O., Marcela Charfuelán, Andrea Corradini, and Manish Mehta: Second prototype version of conversation management and response planning for H.C. Andersen. HLT project NICE Report. NISLab, Denmark, October 2004.