DIALOGUE MANAGEMENT SYSTEMS FOR INTERACTIVE STORYTELLING

DIPLOMA THESIS FOR A BSc DEGREE IN MATHEMATICS AND COMPUTER SCIENCE

Author: Martin Ruskov

Darmstadt, September 2005
CONTENT

Content....................................................................................................................................3
Content....................................................................................................................................3
Content....................................................................................................................................3
Content....................................................................................................................................3
Content....................................................................................................................................3
Content....................................................................................................................................3
Content....................................................................................................................................3
List of figures........................................................................................................................5
List of figures........................................................................................................................5
List of figures........................................................................................................................5
List of figures........................................................................................................................5
List of figures........................................................................................................................5
List of figures........................................................................................................................5
List of figures........................................................................................................................5
1. Introduction..........................................................................................................................6

2. Existing approaches and solutions......................................................................................7
  2.1 Façade.................................................................................................................................8
    2.1.1 Motivation and Prerequisites.......................................................................................8
    2.1.2 Information extraction from utterances........................................................................8
    2.1.3 Decision-taking mechanism.......................................................................................9
    2.1.4 Generation of speech, gestures and emotional expressions.....................................13
  2.2 VICTEC.............................................................................................................................14
    2.2.1 Motivation and Prerequisites.....................................................................................14
    2.2.2 Information extraction from utterances........................................................................14
    2.2.3 Decision-taking mechanism.......................................................................................15
    2.2.4 Generation of speech, gestures and emotional expressions.....................................17
  2.3 NICE Fairytale game system.............................................................................................18
    2.3.1 Motivation and Prerequisites.....................................................................................18
    2.3.2 Information extraction from utterances........................................................................19
    2.3.3 Decision-taking mechanism.......................................................................................19
## 2.3.4 Generation of speech, gestures and emotional expressions

### 2.4 Mission Rehearsal Exercise
- **2.4.1 Motivation and Prerequisites**
- **2.4.2 Information extraction from utterances**
- **2.4.3 Decision-taking mechanism**
- **2.4.4 Generation of speech, gestures and emotional expressions**

### 2.5 NICE H.C. Andersen
- **2.5.1 Motivation and Prerequisites**
- **2.5.2 Information extraction from utterances**
- **2.5.3 Decision-taking mechanism**
- **2.5.4 Generation of speech, gestures and emotional expressions**

## 3. Important aspects and detailed comparison
- **3.1 Common solution**
- **3.2 Structural differences**
- **3.3 Usability tests**

## 4. Resulting Solution
- **4.1 ZGDV Storytelling platform**
- **4.2 The game Ask and Answer**
- **4.3 Proposals for further development**
  - **4.3.1 The place of Ask and Answer in the comparison**
  - **4.3.2 Improvements influenced by the discussed approaches**
  - **4.3.3 Other proposals**

## 5. Conclusion

## 6. References
LIST OF FIGURES

Figure 2.1.1: Discourse act classification in Façade..........................................................9
Figure 2.1.2: Overall architecture of Façade....................................................................10
Figure 2.2.1: General categories of language actions in FearNot!.................................15
Figure 2.2.2: Design of a synthetic character in VICTEC..............................................16
Figure 2.3.1: Dialogue acts in NICE Fairytale World.....................................................19
Figure 2.3.2: Character responses in NICE Fairytale World...........................................22
Figure 2.4.1: Examples of action and resulting state in Mission Rehearsal Exercise....23
Figure 2.4.2: Example for question representation in Mission Rehearsal Exercise..24
Figure 2.4.2: Information state layers in Mission Rehearsal Exercise.........................25
Figure 2.5.1: Architecture of NICE HCA’s Character Module.................................28
Table 3.1: Characteristics of the systems under discussion........................................33
Table 3.1: The place of Ask and Answer among the systems under discussion...........40
Figure 4.3.1: Game frame level of the Ask and Answer transition network..............43
Figure 4.3.2: Question frame level of the Ask and Answer transition network........44
1. **Introduction**

Dialogue-based conversation is the most intuitive way people communicate. That is why it is only natural that efforts are made to enable dialogue driven human-computer interfaces. With the advancement of virtual reality and computer linguistics, it has become possible to create virtual humans - computer-controlled models of that behave and talk like real people. It even becomes possible to experiment the feasibility of theories of cognitive psychology on them.

One of the main fields where virtual humans (called also virtual characters, synthetetic agents or simply avatars) find their application is interactive storytelling. Imagine a story where it is up to you as a consumer or simply a player how interesting the story would become or what would be the end of it - that is interactive storytelling. These applications have without a doubt an entertaining value, but as any story, they can also be educational, thus becoming a form of edutainment. For the purposes of interactive storytelling it is not always important to achieve realism, but rather to create a believable virtual world.

Although for a long time research has been made on spoken dialogue systems, it has been mainly strict for task-oriented applications, whose requirements are quiet different. Also in interactive storytelling (as in general for avatars) dialogue behaviors could be enriched with emotional expression and gestures. This makes communication more realistic and easier to understand. That's why emotional models and other means of articulation sometimes also need to be considered.

The goal of this work is to discuss in detail several state-of-te-art approaches to dialogue management (part 2), as a result I strive to find several points of difference and criteria (part 3) and to suggest decisions for the development of a game, called Ask and Answer, on top of the already existing ZGDV Storytelling Platform (part 4).
2. **Existing approaches and solutions**

The approaches to dialogue management for a long time were mostly task-oriented. These are modeled under the assumption that the system and the user share a common task and thus the user is motivated to make himself clear to the system in order to achieve the goal [Allen et al 01]. Several surveys [Churcher et al 97],[Catizone et al 02], have been made to compare the progress of these systems. Another, less exploited, class, the domain-oriented dialogue managers [Bernsen and Dybkjær 04b], [Gustafson, Lindberg and Lundeberg 99] do not concentrate on a well-defined single task and are close to the domain-restricted version of a Turing test [Turing 50]. Recently, with the emergence of immersive virtual worlds, the field of interactive storytelling has started to attract interest. In these game-like applications, the main dialogue objective has shifted towards the user-aided creation of a storyline, and usually is not something that the user perceives as a task. On the other hand, dialogues in these worlds allow other, non-verbal communication channels, such as gestures and emotional expressions. To meet these new challenges, another style of dialogue management that I call story-oriented has emerged; here the user is viewed as a co-author of a dialogue-centered plot.

In my work I refer to the approaches used in several recent dialogue management systems with embodied talking agents. These approaches range from the story-oriented Façade [Matheas and Stern 05], VICTEC [Aylett et al 04] and NICE Fairy-tale [Gustafson et al 05], [Boye and Gustafson 05] through the task-oriented Mission Rehearsal Exercise [Traum, Marsella and Gratch 04] and to the NICE H.C.Andersen projects [Bernsen and Dybkjær, 04a]. The above systems are being discussed in detail in the following pages. Since they are very different from one-another, I first consider the motivation of each. Then I take a look subsequently at information obtained from user input, mechanism for decision taking and ways of response generation. There are other works [Cavazza and Charles 05] that focus on particular sub-problems of dialogue management, but as they are not complete approaches yet, I do not consider them in this comparison. I start with Façade which is the earliest and most complete of the systems under discussion.
2.1 Façade

2.1.1 Motivation and Prerequisites

According to Chris Crawford “Façade is without a doubt, the best actual working interactive storyworld” [Crawford 04].

In Façade, the user plays the role of a longtime friend of the virtual characters, Grace and Trip, an attractive and materially successful couple in their early thirties. During an evening gettogether at their apartment that quickly turns ugly, the user becomes entangled in the high-conflict dissolution of Grace and Trip’s marriage. The user interaction determines the development and outcome of the posed situation. One of the authors’ goals is to achieve high level of user immersion and believability [Mateas and Stern 02]. They put more weight on story coherence than logical reasoning and information retrieval.

The authors argue that stories like character-oriented kitchen sink dramas, lend themselves better to interactivity than plot-oriented action dramas. That is why character attitudes, levels of self-awareness, and overall tension are designed to be regularly progressing. In Façade the dialogue initiative is in the hands of the virtual characters. This allows expectations in the natural language processing (NLP) about the input and allows the user also not to interact if he wishes.

2.1.2 Information extraction from utterances

The NLP in Façade interprets as input for the dialogue management mainly the pragmatic aspect of the user utterances, classifying them into discourse acts. For example, if the player types “Grace isn’t telling the truth”, the NLP system is responsible for determining that this is a form of criticism, and deciding what reaction Grace and Trip should have to Grace being criticized in the current context.
Figure 2.1.1: Discourse act classification in Façade

Semantics play only a secondary role, as the only such information extracted, is the referred topic and only in the case of the particular discourse act “referTo <topic>”. If the player asked "Can I have a glass of Chablis?", for example the WordNet thesaurus can be used to map "Chablis" to the more general "wine" or even more general "alcoholic beverage." In this way the mapping rules don't need to know anything about "Chablis" (and the hundreds of other words that denote specific alcoholic beverages), only about "alcoholic beverages." Additionally the template language provides support for matching stemmed forms of words, and for matching hierarchical (and recursive) template structures.

Limited gesture articulation by the user (as sipping drinks, kissing etc.) is also supported and interpreted as discourse acts. On the figure are shown the supported discourse acts as stated in [Mateas and Stern 02].

Note that, for example, the “agree” discourse act is not the only discourse act that can map into a “player is agreeing” reaction. In some situations, the “praise” discourse act may be interpreted as agreement; whereas in other situations, “praise” may be considered “not a direct answer”. These Reaction Decider mappings are hand-authored appropriately for the particular situation of the current beat (see later for an explanation on story beats).

2.1.3 Decision-taking mechanism

Architecture and content structure are tightly interdependent and as the authors say “two sides of the same coin” [Mateas and Stern 05]. The high interconnectedness of Façade allows very fine authoring, but makes Façade’s
architecture highly complex and difficult to be reused in other context or domains.

**Drama Manager**

(sequences beats)

**Story Memory**

<table>
<thead>
<tr>
<th>Bag of beats</th>
<th>Desired value arc(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>beat</td>
<td>selected beat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current story values</th>
<th>Previous action time</th>
</tr>
</thead>
<tbody>
<tr>
<td>beat</td>
<td>beat</td>
</tr>
<tr>
<td>beat</td>
<td>beat</td>
</tr>
<tr>
<td>beat</td>
<td>beat</td>
</tr>
</tbody>
</table>

Activity not part of a beat

**Story World**

- Player
- Trip
- Grace

**Natural Language Processing**

- surface text → discourse acts
- discourse acts → reactions

*Figure 2.1.2: Overall architecture of Façade*

The reactions to discourse acts are mostly beat-specific¹, allowing different effect in different situations for the same utterances. Beside these reactions it may take patterns of discourse acts to influence the story-specific variables (social game score). In more detail, the system keeps track of the overall story tension level, which is affected by player moves. For example, a reaction to “praise Grace” may also shift the Player’s affinity towards Grace. Or, a reaction to “referTo infidelity” may increase story tension. Every change in game’s state is performed by the virtual characters in emotionally expressive, dramatic ways.

The *Façade* dialogue management architecture (Figure 2.1.2) consists of characters written in ABL and a Drama Manager that manages the high-level plot, resp. dialogue. In order to integrate a plot, the authors use narrative sequencers, that implement small-scale social games, written in ABL (A Behavior Language), often taking advantage of ABL’s support for reflection in the form of meta-behaviors that can modify the runtime state of other behaviors. ABL is a specially

¹ Façade’s story beats are explained later, for now this could be interpreted as situation-specific
designed reactive-planning language. The Drama Manager, sequences dramatic beats, depending on the previous user interaction and overall story development. These beats are described in a java-written custom drama management language.

A *Facade* beat, the entity that defines the momentarily context, is comprised of anywhere from 10 to 100 *joint dialog behaviors* (jdb's), written in ABL. Each beat is in turn a narrative sequencer, responsible for sequencing a subset of its jdb's in response to player interaction. Only one beat is active at any time. A jdb, *Facade*'s atomic unit of dramatic action consists of a tightly coordinated, dramatic exchange of 1 to 5 lines of dialog between Grace and Trip, typically lasting a few seconds. As the dialogues in Façade are story-oriented, beat's jdb's are organized as a conversation between the virtual characters with common narrative goal, such as a brief conflict about a topic. It is here where the asynchronous autonomous agents Grace and Trip synchronize their behavior and coordinate their dramatic performances – for example, where one speaks and the other reacts with a look and gesture. The user interaction consists of in intervening in these dialogues and thus modifying the plot. Each jdb is capable of changing one or more values of the global story variables, such as the affinity game's spectrum value, or any of the therapy game's self-revelation progression counters, or the overall story tension level.

Two typical uses of jdb's within beats are *beat goals* and *beat mix-ins*. A beat contains a canonical sequence of narrative goals called beat goals. The typical canonical sequence consists of a transition-in goal that provides a narrative transition into the beat (e.g. bringing up a new topic, perhaps connecting it to the previous topic), several body goals that accomplish the beat (in affinity game beats, the body goals establish topic-specific conflicts between Grace and Trip that force the player to choose sides), a wait goal in which the virtual characters wait for the player to respond to the head game established by the beat, and a default transition-out that heads out of the beat in the event of no user interaction. In general, transition-out goals both reveal information and communicate how the user's action within the beat has changed the context and more specifically the affinity dynamic. The canonical beat goal sequence defines how the beat unfolds and, in case of no interaction, ends.

There is also a set of handler *meta-behaviors* that interpret player's
discourse acts, and modify the canonical sequence in response, typically using beat mix-ins. Beat mix-in jdb's are beat-specific reactions used to respond to player actions and connect the interaction back to the canonical sequence. Handlers are responsible both for potentially adding, removing and re-ordering future beat goals, as well as interjecting beat mix-ins into the canonical sequence. Aside beat mix-ins there are *global mix-ins that* lead to satellite topics such as marriage, divorce, sex, therapy and particular objects or such as generic reactions to praise, criticism, flirtations and oppositions. For example, if the player refers to divorce during the “FixDrink” beat, Grace and Trip mix in a short beat goal about their feelings about divorce – and then resume where they left off with their FixDrink beat goals. Or while quibbling about their second honeymoon in Italy, or arguing about what type of drinks Trip should serve (affinity game beats, chosen by the drama manager), it is safe to mix in dialog about, for example, sex, or the wedding photo (hot-button game mix-ins, triggered by a player’s reference to their topics). There are also global mix-ins for misunderstanding or inappropriate input from the player, that try to cover the confusion.

The Drama Manager serves as a beat selection mechanism and is only active when the current beat is finished or is aborted. It is at this level where causal dependence between major events is handled.

In a beat sequencing language the author annotates each beat with selection knowledge consisting of preconditions, weights, weight tests, priorities, priority tests and story value effects. A combination of these is used to determine the context and plot of story continuation, depending mostly on the content authored in the knowledge base. In Façade an target function is the overall tension level, aiming to take the form of an Aristotelian arc. For example, it does not matter which order Grace and Trip argue about Italy, their parents, redecorating, fixing drinks, or their dating anniversary.

A key requirement for making lifelike believable agents is to endow them with the ability do several intelligent activities in parallel – for example, to gaze, speak, walk, use objects, gesture with their hands and convey facial expressions, all at the same time. The selection technology and the fine-granularity of beats, practically allow perceivably parallel addressing of several topics in a similar to Petri-Nets way. Long-term autonomous behaviors, such as fixing drinks and sipping them over time, or carrying around and compulsively playing with an
advice ball toy, last longer than a 60-second beat or a 10-second global mix-in. For example, beside the simple drink sipping, it can be as complex as making drinks and carrying them to the player. If a long-term behavior needs to speak some dialog, it spawns its own beat goal mixins. The characters are enabled to perform extended, coherent series of low-level actions in the background over the course of many minutes, across several beat boundaries, thus realizing another aspect of parallelism. Performance behaviors are often spawn-goaled off to the root of the ABT, which allows the beat goal step itself to finish up and move on to the next beat goal step before the performance behavior is actually done. For example, this allows a character to begin the process of walking somewhere (by spawning off a staging behavior), and then continue on with subsequent lines of dialog while the walking is taking place.

The behavior author may decide that a certain beat goal is important or intense enough that a chosen reaction to a player action should not be immediately responded to, and instead be delayed until the beat goal gist point. For example, if Trip and Grace are yelling at each other, they can believably ignore the player until their yelling is over. Usually upon user interaction and interruption of a beat goal, the newly inserted beat goal immediately begins executing, and later on the previous aborted one will re-run if it hadn’t already reached its gist point. For example, in the “FixDrinks” beat, Grace and Trip are ready to finish up this beat once they’ve done a sufficient amount of quibbling over what they think the player should drink. Short, alternate uninterruptible dialog is authored for each beat goal for that purpose. Also, each beat goal has a reestablish jdb that gets performed if returning to the beat from a global mix-in.

With only a few exceptions, the narratives of affinity game beats themselves are also designed to be causally independent of one another, relating to the sparse plot.

2.1.4 Generation of speech, gestures and emotional expressions

The generated output is hard-coded sentences with most jdb's authored with three to five alternates for expressing their narrative content at different combinations of player affinity and tension level. These include variations in word choice, voice-acting, emotion, gesture, and appropriate variation of information revealed.
2.2 VICTEC

2.2.1 Motivation and Prerequisites

The EU framework V project VICTEC - Virtual ICT (Information and Communication Technologies) with Empathic Agents - project aims to develop a system for social education for children, and more particularly bullying. Its FearNot! Demonstrator is the work-in-progress implementation of the emergent narrative theory [Aylett et al 04]. The authors pose three features in FearNot!, that allow for simplification of the NLU:

1. The dialog is intended to be very short and focused only on the previous bullying.
2. The intended users are children of age 10 and so are supposed only to type simple sentences.
3. In order to help the children buttons providing part formed sentences are intended to be provided.

Another important aspect in the design is that the approach must also take account of cross-cultural language practices such as the specific language used in schools in the UK, Portugal and Germany, the countries of the project partners. As a result language actions\(^2\) are used in FearNot! both for interaction between the virtual characters and with the user.

2.2.2 Information extraction from utterances

In VICTEC as in Façade, the NLU (natural language understanding) uses syntactical matching on the keywords included in it, and an attempt is made to keep the conversational initiative with the agent rather than the user. The authors make use of the expectation driven dialogue, anticipating certain types of answers from the user and comparing the chosen answer with a set of pre-defined templates. In this way, they hope to achieve more believable conversational behavior of the system.

The authors argue that speech acts (as defined by Searle [Searle 69]) are not suitable enough, because of their very high level of abstraction and the fact that only a subset of those generally used are relevant to bullying scenarios. Another set of appropriate actions for bullying and victimization interactive scenarios has been identified. In order to include semantics and to reduce ambiguity, the authors have added some semantic information to the agents

---

\(^2\)Language actions are explained later, for now this could be interpreted as a form of speech acts
actions. They call the combinations of speech act plus semantic information language actions and classify them in three categories: Help, Confrontation and Socializing (figure).

To solve the problem of how to identify sentences with language actions, the authors intend to apply microgrammars to the very small set of sentences that have been classified in the knowledge base. They note that a microgrammar can be written for each language action. Microgrammars comprise sets of features which are classified into three different types:

- Words and collocation: certain words and particularly combinations of words (collocation) indicate some speech acts. For example the words `who, when, where’, indicate questions.
- Prosody: the tone of voice used in an utterance may indicate its intended act. In English, questions, for example, can be indicated by a rising intonation at the end of a sentence.
- Conversational Structure: the current context and the immediate predecessor statements may give an indication of the speech act. A simple example of which is that the utterance after a question is probably a reply or a request for clarification.

<table>
<thead>
<tr>
<th>Category</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFRONTATION</td>
<td></td>
</tr>
<tr>
<td>SOCIALISING</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 2.2.1: General categories of language actions in FearNot!*

### 2.2.3 Decision-taking mechanism

FearNot! draws on techniques and technologies inspired by research in conversational agents [Braun 02], [Rist et al 03], [Prendinger and Ishizuka 01]. A deep appraisal of events and the other characters is carried out, using the
emotion-modelling system of Ortony, Clore and Collins [Ortony et al 88] and the resulting emotional state is combined with the character’s goals and motivations to select an appropriate action.

The authors describe an action as a collection of the following instances:

- an object on which the action can be performed (an object in the environment or another character)
- the agent performing the action
- the action priority (used to order and deal with conflicting actions)
- the context in which the action is performed (i.e. location, props, internal goal, history of previous actions, topics)
- the emotional status of the character at that time, and the utterance, if any, relating to language actions, that should be played
- the animation of the body of the character involved and accompanying gestures.

![Figure 2.2.2: Design of a synthetic character in VICTEC](image)

The emotional status of the character feeding into its action selection mechanism will determine whether the action to be performed is implemented via language action, physical activity or both (figure 2.2.2). A Finite State Machine (FSM) approach is taken to limit the number of language actions from which the
action-selection mechanism can choose.

In the case of agent-to-agent communication, there is no ambiguity about the action – whether physical or language – that has been selected by one character to which a second character has to respond. In the case of a language action, there is also no ambiguity about the content of the utterance for which a response must be generated, and since this also comes from the templates database, it is much less demanding than responding to an unlimited repertoire from a human user.

In the child-to-agent case, the problem of dealing with unexpected inputs still exists. Here, the FearNot! demonstrator would specifically drive the conversation by using leading questions with a limited range of options for answer. The question-asking strategy is one used originally by ELIZA [Weizenbaum 66], but rather than the more open domain of an individual’s emotional problems, FearNot! covers the much more specific one of bullying in the strong context of the episode the child has just seen. The authors believe that this, together with the provision of ‘sentence starters’, to help slow typers, will allow the system to behave competently. In child-to-character interaction the previous language action is not known, but must be inferred. The incoming text is matched against a set of language templates, and the language and action index is then taken as the starting point for the language action with which the agent must respond. Since an objective is to retain control of this dialogue by keeping the conversational initiative with the character, FSM could also be used to generate expectations about what language actions the child has produced. Each action category possesses its own organization and consequently requires the design of its own FSM. Each FSM integrates the language actions relative to the category itself but also potential elements of answers for discussion or interaction.

2.2.4 Generation of speech, gestures and emotional expressions

In VICTEC the response generation is also managed with language acts. However these do not uniquely specify the utterance in which they are expressed. When parsed in conjunction with the semantic information and contextual knowledge of the source and sender of the speech act, the above mentioned microgrammars should be able to generate a sentence [Aylett et al 04]. In addition,
language and physical actions must form coherent sequences, accepted as such by the child users. In order to generate the utterance for a selected language action, it has been decided to use a modified shallow-processing approach, based on that originally used in ELIZA and more recently in chat bots [Mauldin 94]. Thus the template database typical of such systems has been extended by indexing it via language actions.

In the case of an agent-to-agent communication, the process starts with the selection of a language action and ends with the selection of an utterance.

The opposite occurs in the case of agent-to-user communication since the system needs to recognize an utterance via keywords and then select an appropriate language action or action in order to provide an answer to the user. Those actions can be triggered and generate agent utterances according to their emotional states.

For example consider the act of greeting a person. The set of possible sentences is very small, consisting of a greeting word, possibly the name of the person being greeted, and possibly a greeting question.

- Hello
- Hello Sue
- Hi
- Hi Tom
- Hi Jo, how are you?

There could be seen a general form to these sentences which can be expressed in Backus Naur Form (BNF) as follows:

```
<Greeting word> <ToName><status_question>
```

```SFML
<Greeting word> = 'Hello' | 'Hi'
<ToName> = < receiver>
<status_question> = 'how are you' | 'are you all right'
```

Here, the term <receiver> is a context variable that is set by the semantic information in the language act.

### 2.3 NICE Fairytale game system

#### 2.3.1 Motivation and Prerequisites

In the NICE fairy-tale system an approach common to games is used. There are two main characters: the helper Cloddy Hans, introduced to facilitate
progression through the story and gatekeeper Karen, introduced as an obstacle in the story.

2.3.2 Information extraction from utterances

User interpreted utterances are represented by tree-structured dialogue acts. A more detailed description is to be published [Boye et al 06]. The NLU output can be categorized as in the figure.

Instructions: "Go to the drawbridge", "Pick it up"…
Domain questions: "What is that red object?", "How old are you?"…
Giving information: "I'm fourteen years old"…
Negotiating utterances: "What do you want in return?", "I can give you the ruby if you lower the bridge"…
Confirmations: "Yes please!", "Ok, do that"…
Disconfirmations: "No!", "Stop!", "I didn't say that!"…
Problem reports and requests for help: "Help!", "What can I do?", "Do you hear me?"…
Requests for explanation: "Why did you say that?", "Why are you doing this?"…

Figure 2.3.1: Dialogue acts in NICE Fairytale World

2.3.3 Decision-taking mechanism

The dialogue management software consists of a kernel laying down the common functionality, and scripting code modifying the dialogue behavior as to be suitable for different characters and different situations. This model of code organization is common in computer games [Varanese and LaMothe 03]. In this approach plot milestones are defined as important points of story development. Such a milestone denotes a irreversible point in time that changes the story context. Such milestones are for example, Cloddy Hans picking the diamond or Karen opening the gate.

The dialogue management kernel issues dialogue events at important points in the processing [Gustafson et al 05]. Some kinds of dialogue events, the so-called external events, are triggered from an event in a module outside the dialogue manager. These include:

- BroadcastEvent (some other character has said and done something)
- GestureEvent (the Gesture Interpreter has recognized a gesture)
- ParserEvent (the natural language parser has arrived at an analysis of the latest utterance)
- PerformedEvent (the animation system has completed an operation)
- RecognitionFailureEvent (the speech recognizer has detected that the user has said something, but failed to recognize it)
- WorldEvent (an event has occurred in the 3D world)
- TriggerEvent (the animation system has detected that the character has moved into a trigger)

There is one type of external dialogue event for each input message that the dialogue manager can receive.

Another category of events, the internal events, take place within the dialogue kernel. Examples are:
- AlreadySatisfiedEvent (a goal which already is satisfied has been added to the character’s agenda)
- CannotSolveEvent (an unsolvable goal has been added to the character’s agenda)
- IntentionEvent (the character has an intention to say or do something)
- NoReactionEvent (the character has nothing on the agenda)
- PossibleGoal-ConflictEvent (a goal is added to the agenda, but the agenda contains a possibly conflicting goal)
- TimeOutEvent (a timeout has expired)

Others such as RequestEvent, QuestionEvent are also generated as an effect of specific dialogue acts made by the user (e.g. Request, Question). The kernel provides a number of operations through which the scripting code can influence the dialogue behaviour of the character [Boye and Gustafson 05]. In the NICE fairy-tale world these are:

1. interpret an utterance in its context
2. convey a dialogue act
3. perform an action
4. add a goal to the character's agenda
5. remove a goal from the character's agenda
6. find the next goal on the agenda
7. pursue a goal on the agenda.

The perform operation ultimately leads to an action being performed by the animated character. For instance, consider the case where the user requests Cloddy Hans to "Go to the fairy-tale machine". This would lead to the following sequence of events:

1. A message from the parser arrives and generates a ParserEvent.
2. The ParserEvent is caught by the scripting code of the current scene, which calls the content interpretation procedure of the dialogue kernel.
3. Content interpretation establishes that the user's utterance is a request from the user for Cloddy Hans to go to a specific spot (the fairy-tale machine, in this case). A RequestEvent is generated.
4. The RequestEvent is caught by the scripting code, which calls convey to produce an utterance acknowledging the request, and then adds to Cloddy Hans's agenda the goal that he should be standing next to the fairy-tale machine.
5. When Cloddy Hans has eventually reached his destination, a message arrives from the animation system. This message generates a PerformedEvent, which can again be caught to produce a new utterance from Cloddy Hans, etc.

For instance, in the example above, more input from the user may arrive when Cloddy Hans is walking over to the fairy-tale machine. Using the event-based model outlined above, that is no problem; a new line of dialogue can be opened and the user's new utterance can be answered. Eventually the PerformedEvent in (5) above will arrive, and Cloddy Hans can then be made to switch back to the original line of dialogue. The interplay between the instructions in the scripting code and the dialogue events generated by the dialogue management kernel creates the overall dialogue behaviour of the character.

As an example, if the speech recognizer detects that the user is speaking but cannot recognize any words, it sends a “recognition failure” message to the dialogue manager. The dialogue management kernel receives this message,
generates a RecognitionFailureEvent, and calls the onDialogueEvent procedure of the current scene. The current scene may then re-direct the procedure call to its current phase. In this way, different pieces of scripting code can be provided for different characters, scenes and phases, facilitating the creation of different personalities and scene-dependent behaviour in a modular systematic way. Dialogue events can be caught by the scripting code by use of a callback procedure.

This event-driven model allows for asynchronous dialogue behaviour (i.e. a character in the fairy-tale system is not confined to a model where the user and character have to speak in alternation). Rather, a character may take the turn and start speaking for a number of different reasons: because the user has said something, because some other fairy-tale character has said or done something, because of an event in the fairy-tale world, or because a certain amount of time has elapsed. Such events arrive asynchronously; hence they give rise to a more flexible dialogue model.

2.3.4 Generation of speech, gestures and emotional expressions

**Responses to instructions:** either accepting them ("OK, I'll do that") or rejecting them, ("No I won't open the drawbridge!", "The knife already is in the machine").

**Answers to questions:** "The ruby is red", "The knife is on the shelf", "I am 30 years old"…

**Stating intentions:** "I'm going to the drawbridge now"…

**Confirmation questions:** to check that the system has got it right, e.g. "So you want me to go to the shelf?"

**Clarification questions:** when the system has incomplete information, e.g. "Where do you want me to go?", "What should I put on the shelf?"…

**Suggestions:** for future courses of action, e.g. "Perhaps we should go over to the drawbridge now?".

**Negotiating utterances:** "I won't do that for nothing", "What a piece of junk! Find something better"…

**Explanations:** "Because I want the axe in the machine".

*Figure 2.3.2: Character responses in NICE Fairytale World*

The convey kernel operation for scripting ultimately leads to an utterance with accompanying gestures from the character (via text generation, graphics generation and speech synthesis). The character responses are classified as shown in the figure.
2.4 Mission Rehearsal Exercise

2.4.1 Motivation and Prerequisites

The Mission Rehearsal Exercise (MRE) uses task-oriented dialogue modeling, and this allows a deeper semantics understanding. It follows the information state approach, defined in the Trindi project [Larsson and Traum 03]. Information state is the part of the context deemed relevant for dialogue modeling and serves as a snapshot of the dialogue state. It is being updated by dialogue moves, the abstract description of the spoken interface.

2.4.2 Information extraction from utterances

(1) agent sgt
event secure
patient assembly-area
type act

(2) object-id assembly-area
attribute safety
value secure
polarity positive
type state

Figure 2.4.1: Examples of action and resulting state in Mission Rehearsal Exercise

The core utterances could influence the topic under discussion or establish or resolve commitments and obligations of conversational participants towards states and actions. In MRE their content could be interpreted as a state or action description or a question about some of these. In the task model, all states and actions are annotated with semantic information that would match the description of semantically interpreted utterances. In the figure is an example of action of the sergeant securing the assembly area (which can be accomplished by having the squad leaders each secure a quadrant) and the resulting state of the assembly area being secure. The NLP generates similar descriptions from spoken utterances.

As of the date of writing of [Traum, Marsella and Gratch 04] modeled speech acts included the forward-looking assert, info-request, order, request and suggest.
Assertions are supposed to establish a commitment by the speaker that a state holds or that an action happened, is happening, will happen or should happen, depending on tense and aspect of the utterance.

Info-requests represent questions as (possibly partial) propositions with a designated q-slot indicating the questioned part of the proposition. For example, on the figure is shown an info-request by the lieutenant to the sergeant with the content being a question about whether the assembly area is secure.

(3) action info-req
actor lt
addressee sgt
type csa
content q-slot polarity
type question
prop object-id assembly-area
attribute safety
value secure
time present
type state

Figure 2.4.2: Example for question representation in Mission Rehearsal Exercise

There are also several backward-looking acts that include acceptance and rejection.

2.4.3 Decision-taking mechanism

Dialogue acts are used to update the Information State. Decisions of how to act in dialogue are tightly coupled with other action selection decisions in the agent. The information state is also used as context for these aspects of reasoning. He can choose to speak, choose to listen, choose to act related to a task, etc. Aspects of the information state provide motivations to speak, including answering questions, negotiating with respect to a request or order, giving feedback of understanding (acknowledgements, repairs, and repair requests), and making suggestions and issuing orders, when appropriate according to the task model [Traum et al 03b].
The information state is partitioned into five main layers, representing distinct aspects and shown in the figure. A detailed description can be found in [Traum, Marsella and Gratch 04].

Speech acts do not cause direct effect on the beliefs, desires or intentions of the conversational participants and utterances could be insincere. Rather, their direct effect involves social commitments and one could infer from these commitments the beliefs and intentions commonly associated with the utterances under additional assumptions.

The dialogue management in MRE is not event-centered and contains two distinct and interleavable “cycles”, one for NLU and information state update and one for response generation.

The formal representation of the emotional state of the virtual characters is connected to the knowledge base and thus they are able to talk about their emotions [Muller et al 04]. The Emotion and Adaptation (EMA) model, based on [Smith and Lazarus 90] provides a deep process model of the mechanisms underlying emotion, including the cognitive assessments that precede it, their consequence in cognition and behaviour and the way these consequence impact subsequent assessments. These formal representations of dialogue semantics, beliefs and emotions are planned to interact tightly, for example in emotionally resolving ambiguous questions [Traum, Marsella and Gratch 04]. Several example situations follow:

1. If the injured boy’s mother believes the soldiers are abandoning him, this is appraised as a threat to her goals and leads to distress or anger.
2. The mother might convince herself that the troops are going for help, despite perceptual evidence.
3. A threat to a sub-goal (such as the landing zone being secure) might be distressing, not because the sub-goal is intrinsically important, but because it facilitates a larger goal (such as evacuating the boy).

4. When the lieutenant asks “What happened here?” the sergeant agent might openly express true appraisal-based feelings of guilt and concern, for example through facial expressions, gestures, posture, gaze and head movements. However, if the sergeant is coping by shifting responsibility for the accident to the mother, the initial expression of guilt will be quickly suppressed. Instead, the sergeant will express anger at the mother, nonverbally through an angry facial expression, a shaking of his head and a dismissive wave of his hand toward the mother, as well as verbally by saying “they rammed into us”, all in a more calculated attempt to persuade the lieutenant.

2.4.4 Generation of speech, gestures and emotional expressions

Interpretation at different layers is also independent and thus one communicative action could correspond to several parallel acts in separate layers. Generation on the other hand results mainly from a “main-act”, but one utterance could correspond to several acts, (e.g. including turn-taking).

2.5 NICE H.C.Ansersen

2.5.1 Motivation and Prerequisites

An entirely different, domain-oriented approach is taken in the NICE H.C.Ansersen (HCA) system. The authors define a set of principles of prototypical successful human-human conversation [Bernsen and Dybkjær 04b]:

1. Initially, in a polite and friendly way, the interlocutors search for common ground, such as basic personal information, shared interests, shared knowledge, and similarity of character and personality, to be pursued in the conversation

2. The conversation is successful to the extent that the interlocutors find enough common ground to want to continue the conversation

3. The interlocutors provide, by and large, symmetrical contributions to the conversation, for instance by taking turns in acting as experts in different domains of common interest, so that one partner does not end up in the role of
passive hearer/spectator, like, for instance, the novice who is being educated or
trained by the other(s).

4. To a significant extent, the conversation is characterized by the participants
taking turns in telling stories, such as descriptions of personal experiences,
observations, anecdotes, descriptions of items within their domains of expertise,
possibly jokes, etc.

5. Conversation is rhapsodic, i.e. highly tolerant to digression, the introduction of
new topics before the current topic has been exhausted, etc. Yet conversation
also requires a reasonable amount of conversational control and coherence in
order not to fall apart into disjoined semi-monologues.

6. Conversation, when successful, leaves the partners with a sense that it has been
worthwhile.

In these principles entertainment is not mentioned, partly because the
authors assume that successful conversation is entertaining in itself.

2.5.2 Information extraction from utterances

Aside natural language understanding, NICE HCA supports as input also
gesture interpretation, using a touchscreen. Output from the NLU and gesture
interpretation modules reach the Character Module through the Input Fusion
module (figure 2.5.1). The first NICE HCA prototype only performs limited gesture
and combined speech-gesture input processing [Bernsen and Dybkjær 04b].
Essentially, the Input Fusion module simply combines an n-best gesture
interpretation frame from the Gesture Interpreter, if any, with the one-best topic
and domain frame from the NLU module. The combined frame is sent to the
Character Module which performs basic input fusion, when needed.
2.5.3 Decision-taking mechanism

To keep up with the posed principles, HCA asks questions about the user, such as name, age, gender, nationality and opinions on his fairytales, his visible persona, favorite games, and his study. These HCA initiatives serve the goal of conversational symmetry and user contribution. HCA does not just answer questions, or ask them, but tells stories, anecdote and personal views. His topics include his life, his fairytales, wall pictures and other objects in his room. The problem that the system cannot always pursue in depth a topic is solved by either uttering a context-free quote of the real H.C. Andersen or non-situated jokes, or asking a question on different topic or domain, rather than replying [Bernsen et al 04]. HCA replies to insults by also insulting the user, and if he fails to understand the user he may decide to simply talk about something else.

The system has three behavioural states – idle (non-communicative action), listening (communicative function) and speaking (communicative action). The Communicative Function state is elicited as soon as the user starts speaking to or gesturing towards HCA, but it is not influenced by the exact actions of the user. Simultaneously the HCA Character Module is processing this input. When the processing is finished, HCA goes into the speaking or Communicative Action
The behaviour state transitions are managed by the Character Module (CM) Manager which also takes care of module-external communication. User input processing is done by the Character Module’s Mind State Agent (MSA) in consultation with the Knowledge Base (figure 2.5.1). MSA processing is controlled by the MSA Manager. The MSA Manager receives the input sent by the Input Fusion module to the CM Manager wrapped in an XML frame, adapted for internal use in the processing by the MSA modules. The frame incrementally gets filled with various information needed by some of the MSA modules and, eventually, with the output references to be sent to the Response Generator. Thus, the Response Generator receives user’s enquiries together with the response to be expressed.

The conversation could be characterized by its rhapsodic nature (it can go anywhere), by a requirement for a substantial minimum of conversational coherence and by the absence of task constraints. Therefore the system needs, first of all, a general grasp of the conversation in terms of conversational memory, and on the other hand a mechanism for continuation, including whether or not to respond to the user’s input and how to possibly continue the conversation.

Essentially, the Input Fusion module simply combines an n-best gesture interpretation frame from the Gesture Interpreter, if any, with the one-best frame from the NLU module. The conversation control responsibilities are packed in the Conversation Intention Planner (CIP). Apart from looking at the input topic and domain determined by the Input Fusion, the CIP performs no processing of the user’s input. These are also the only semantic information, extracted by the NLU. Other processing is left primarily to the Domain Agents. The domain and topic information is used by the CIP to decide and propose how to continue the conversation. A CIP-proposed continuation may be a question, a statement, or a mini-dialogue. A question has the function of maintaining HCA’s control of the conversation. A statement offers an opportunity for the user to seize initiative. A mini-dialogue is a predefined small task-oriented spoken dialogue-style, that will allow HCA, on occasions at which he takes particular interest in the user’s input, to carry out in-depth conversation on a certain topic.

The decision for continuation is based on HCA’s conversation agenda as represented in the CIP as a partially ordered preference of topics that could be discussed only once. The Knowledge Base contains also rules for when to
change domain or topic. In HCA domain changes are basically only made when a domain has been reasonably addressed already, while topics could also be changed in order to hide misunderstandings. On HCA’s conversation agenda, a special, meta domain always takes priority because HCA wants to resolve any meta-communication issues before proceeding. The user domain has also high priority, so the user could introduce himself early on in the conversation. The information collected about him is stored in the User Model. Some output selection rules consider the user’s age. HCA has also some other favourite domains and topics which he will go to some length to pursue in conversation, such as his fairytales.

The Mind State Agent Manager (MSAM) receives in a frame the CIP’s decision concerning whether to respond to the user’s input as well as the CIP’s proposed continuation. If no reply to the user’s input is to be found in the Knowledge Base, the MSAM will always use the continuation. Otherwise, the MSAM will first try to retrieve a reply from the Domain Agents (DAs). If the reply is empty, it will use the continuation. If the reply is non-empty, the MSAM will randomly decide whether or not to use the continuation. To identify a reply and to find the details about a continuation, the MSAM must contact the DAs. For these purposes functions that ask the Knowledge Base are used: \textit{get reply}, \textit{get continuation} and a function with a parameter action is used that indicates the particular meta case to be handled, e.g., \textit{repeat} or \textit{insult}.

The DA component will internally direct the request from the MSAM to the relevant DA. The DAs perform the domain reasoning needed and handle input semantics processing in one of two ways. The user’s input is either sent to the Knowledge Base as an SQL query or – only in case of a mini-dialogue - to the Mini-Dialogue (MD) Processor. The Mini-Dialogue Processor is a finite-state machine, which processes the input in the dialogue context and produces an internal identifier, which the DA uses to query the Knowledge Base and retrieve a reference to HCA’s output. In addition to the CIP’s maintenance of a record of which continuations have been used already, a Conversation History (CH) is accessible to all main Mind State Agent modules. It keeps track of information about the individual input and output turns in the conversation, for instance in order to enable HCA to repeat what he just said, the number of consecutive turns involving meta-communication, and the mini-dialogue status (started, ongoing,
ended). Keeping track of numbers of turns of type $T(n)$ enables the CIP to take specific action when, e.g., the user has produced several low-confidence score input turns. HCA’s emotional state changes as a function of the user’s input, for instance if the user insults HCA, wants to know his age, or shows a keen interest in the Ugly Duckling. In this emotional model, when HCA expresses anger, the anger sometimes diminishes as a result.

2.5.4 Generation of speech, gestures and emotional expressions

Predefined output of HCA includes spoken output based on some 300 different spoken output templates; some 100 different elementary non-verbal behaviours, such as eye blinks, smiles, pointing gestures, or walk-arounds; and spoken/non-verbal output concerning some 20 objects in his study.
3. **IMPORTANT ASPECTS AND DETAILED COMPARISON**

3.1 **Common solution**

In all discussed systems it has been agreed that alternated turn-taking as in task-oriented systems and chatbots is inappropriate. Interactive storytelling involves an *asynchronous conversation* and requires some form of reasoning to determine when to reply verbally, to continue the dialogue in some other manner or just passively accept user utterances. When the user does not interact, it is also common to these systems that they initiate further evolvement of the situation.

3.2 **Structural differenced**

The *orientedness of the dialogue* is the way systems approach to a dialogue. MRE uses the well founded task-oriented approach, as its system uses a concept similar to TrindiKit [Larsson and Traum 03]. Task-orientedness is characterized by a common task between the user and the system, so both sides are interested in successful communication that would solve the task. NICE HCA uses a domain-oriented approach, its characteristics are described in Section 3.5.1. Although not clearly stated, VICTEC is probably also intended to be organized in a similar way. As for NICE Fairytale and Façade, they could also be considered as domain-oriented systems. However, upon topic change, the characters attempt to get the conversation back to the intended plot as soon as possible, that is why I find it necessary to distinguish these from domain-oriented systems and call them story-oriented.

Two *dialogue initiative approaches* are used in storytelling systems. The system-led conversation, as one of the options, puts the dialogue initiative in the hands of the system. Thus the approach allows anticipation of what the user could respond and simplifies the work on understanding it and adequately continuing the talk. This approach is taken in Façade and VICTEC. Similar ideas have been considered also in NICE and MRE, but there the systems aim at a mixed user and agent initiative, depending on short-term goals. In task-oriented dialogue systems also user-initiated dialogues are used. However for non-task-oriented systems, this is much more challenging because of the absence of a common task. That is the reason why so far user-led dialogues have been
avoided in interactive storytelling.

<table>
<thead>
<tr>
<th></th>
<th>Façade</th>
<th>VICTEC</th>
<th>NICE Fairytale World</th>
<th>Mission Rehearsal Exercise</th>
<th>NICE H. C. Andersen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientedness approach</td>
<td>story</td>
<td>domain</td>
<td>story</td>
<td>task</td>
<td>domain</td>
</tr>
<tr>
<td>Dialogue initiative</td>
<td>system</td>
<td>system</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Number of speech acts</td>
<td>36</td>
<td>3</td>
<td>categories</td>
<td>8 ...</td>
<td>7+</td>
</tr>
<tr>
<td>Semantics extraction</td>
<td>topic (alternative to pragmatics)</td>
<td>integrated in pragmatics</td>
<td>N/A</td>
<td>Deep form-filling approach</td>
<td>topic, domain</td>
</tr>
<tr>
<td>Misunderstanding handling</td>
<td>concealed</td>
<td>N/A</td>
<td>N/A</td>
<td>repaired</td>
<td>ignored</td>
</tr>
<tr>
<td>User personality</td>
<td>artificial</td>
<td>N/A</td>
<td>artificial</td>
<td>N/A</td>
<td>HCA asks questions about the user</td>
</tr>
<tr>
<td>Emotional model</td>
<td>none</td>
<td>Ortony et al</td>
<td>none</td>
<td>EMA, based on Smith and Lazarus</td>
<td>own simplistic</td>
</tr>
<tr>
<td>Dialogue memory management</td>
<td>partially ordered non-replayable beats</td>
<td>N/A</td>
<td>plot milestones</td>
<td>information state</td>
<td>previous input and output turns; covered topics</td>
</tr>
<tr>
<td>Gesture generation</td>
<td>hard-coded in content</td>
<td>physical act selection</td>
<td>N/A</td>
<td>deep character intention model</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Table 3.1: Characteristics of the systems under discussion*

The **pragmatic interpretation of utterances** takes a crucial place in the overall dialogue coherence of all systems. Different approaches have been taken to classify user interaction into a set of dialogue acts. However in contrast to recent works in dialogue act theory [Stolcke et al. 00], the act sets chosen by storytelling systems are adapted to match the application context and purpose. However the exact set of acts is publicly listed only for Façade.

As for **semantic information**, most systems are limited to topic and/or domain retrieval (see table). Although not exactly formulated, presumably in VICTEC an approach similar to the one in Façade is taken. An exception makes MRE: There an attempt is made to extract complete semantic meaning of an utterance, modeling the complete situation, this is made with an form-filling approach similar to those commonly used in task-oriented systems.
As for the question how to deal with *unclear or wrong user input*, different approaches have been taken, according to the specific scenario. Most effective is Façade’s and NICE HCA’s concealing of the failure of the NLU by acting as if the content of the misunderstood utterance is unimportant. A deeper processing approach has been used in MRE. There misunderstood utterances are being grounded by requesting repairing information following [Traum 94]. In the other systems, the problem has been left open, as for example in the NICE Fairytale world it has to be handled by authoring scripts, depending on the exact situation.

The only system that tries to collect *information about the user* is HCA. In MRE, because of the specific application area this is not considered. The story-oriented systems define themselves out of the issue by giving the user an artificial personality. In VICTEC, so far, this issue is not addressed in the scientific papers.

Different *emotional models* have been used. While VICTEC and MRE use theoretical deep-processing models, in HCA a simple own model is being developed. Façade and the NICE Fairytale world take a more pragmatic approach and straightforward hard-coded emotions are implemented into the story content.

The *dialogue memory management* is the way the system keeps track of what has happened so far and uses this to handle repetitions. This includes deciding how to react to the user repeating a statement, as well as asking for a repetition of some previous system utterance. Façade and the NICE Fairytale world avoid the problem, by allowing only straightforward dialogue evolvement and authoring content to handle particular situations. In Façade the beats used are non-replayable once their gist point is reached. In NICE Fairytale the plot milestones are points of irreversible change. No special works on repetitions is released for VICTEC and MRE yet. HCA keeps track of the passed dialogue and can answer to repetition requests. The effect of the domain priority selection used in HCA is similar to the state sequencing used in Façade and NICE Fairytale.

*Gesture generation* is another issue in which two different approaches are taken. In Façade it is up only to the author to formulate gesture articulation inside the expressed content. In VICTEC and HCA (in the latter the mechanism is not described) gestures are chosen separately from speech replies.
3.3 Usability tests

As the systems under discussion pursue very different goals, their usability tests also focus on different aspects. For NICE, test results (Fairytale [Gustafson et al 05] and HCA [Bernsen and Dybkjær 04c], [Bernsen et al 04]) have been released. The VICTEC and MRE are currently being developed and there are no usability results. Façade is free for download and personal experience.
4. Resulting Solution

4.1 ZGDV Storytelling Platform

The ZGDV storytelling platform has principle functionality for dialogue management, however complete story content for it has only been authored for user gesture interaction and sentence-based interaction has not been applied in practice yet.

The Dialogue Manager component contains a recursive transition network which employs the W3C XML-based functional language XEXPR\(^3\) [Nicol 00]. XEXPR allows defining and accessing global variables, so that story development can easily be made dependent of previous interaction history. This transition network is called recursive, because each of its states, could contain a network in itself. To keep it simple, I call such states with the name frames, to distinguish them from the simple, finest-level states.

Generally one could consider simple states in the transition network as points of interaction, thus not connecting them to a theoretical entity as made in Façade [Mateas and Stern 05]. On the other hand these theoretical entities could be represented by frames. The unlimited levels of nesting, allow the modelling of even several levels of story entities as chapters, levels, scenes, dramatic beats, etc. This leaves the choice of model and granularity of the story to the autor and allows the definition of new levels of abstraction, motivated by storytelling theories.

Transitions are guarded by XEXPR constraints or Jess rules. Besides logical and control functions, XEXPR supports functionality for dialogue acts as recursive pattern matching [Wallace 00] and regular expressions. So far support for these in Jess is not implemented. The constraints are checked against user-generated or internal system events.

System actions are written in XEXPR, nesting a sequencing XML-based language PlayerML to generate character reactions or changes in the virtual environment.

The Storytelling Platform supports the following functionality:

- a gradual addition to rule based behavior to the transition

---

\(^3\) XEXPR (for eXtended EXPression) as an abbreviation has been introduced by ZGDV e.V. and is not mentioned in the original document
network
- encapsulates a certain behavior from its implementation either as a transition network or as a rule system. (Every behavior element can be implemented either way, without need to change super-frames or elements of sub-frames).
- enables the controlled use and activation of rule subsets from within the story environment; the scope of the rules validity is equally controlled and limited.
- enables the hierarchical composition of complex behaviors out of dependent behaviors.

The XEXPR functional language in dialogueML is going to be extended with ontology binding functions (queries).

In the ZGDV Storytelling Platform architecture several different modeling components are being encapsulated. This is done for reasons, comparable with the motivation for the Information State approach in TrindiKit [Traum and Larsson 03]. In the Storytelling Platform these include: dialogue events, contextual information manipulators, state update mechanism and state selection mechanism. A brief explanation of each of these follows.

There are two types of dialogue events – internal and external (cp. NICE Fairytale world). Internal events are those generated by the system. An example for such event is TimeoutEvent. External events are generated by the user. Those include coming or going to the scene, utterances and pointing behaviors. Recognition of user facial expressions is on the future work list too.

The contextual information is the domain-specific content of the story. In the development process of the Storytelling Platform, the goal is to make content as easy to author and reuse as possible and make use of sources, independent of the implementation (e.g. online databases).

State update mechanism is the way to decide which states present prospective continuation of the dialogue. This component is built on top of the contextual information, as it uses the knowledge base (e.g. story variables, ontology, etc.) as a criterion for its decisions. As already mentioned, so far XEXPR constraints and Jess rules have been in use.

At the end of the state update process, several new states could be marked as prospective. The state selection mechanism is the one that decides which of
these should be chosen as a continuation. So far only deterministic strategies have been used, taking either the most local (the answer in the exact situation) or the most global (if possible, change of domain, topic, etc) option.

The Storytelling Platform supports authored, as well as automated gesture generation. In the automated case, gestures are inferred from a simplistic analysis of the punctuation of the authored utterance.

In ZGDV, the aim is to create a powerful and yet easy to author storytelling platform, that gradually evolves from shallow expectation-driven interactive dialogues to autonomous virtual humans with deep-processing reasoning and cognitive models.

4.2 The game Ask and Answer

I make this comparison in the scope of a particular game and seek what and how of each approach could be reused or at least influence in its development. Ask and Answer is a competitive, educational, novel multi-user edutainment game that involves two teams of human users and a single virtual character. Within this game, it pays off for the users to create social bounds with the single virtual character, and to adapt their typed input to the limited NLP faculties of the virtual agent [Iurgel and Ziegler 05].

In the game the players are expected to engage in a conversation and persuade the Capricious Non-Player Character (CNPC) to help them solve a TV-styled quiz, rather than simply think of the correct answers. This includes efforts to adjust to his limited natural language understanding capabilities and speaking as clear to it as possible. In this way, the emphasis is being lifted from NLU and focuses on dialogue management and pragmatic pseudo-psychological interaction. The CNPC himself would put all his efforts in helping his preferred team (which could change in the course of a game) and simultaneously try to keep the game as balanced and thus long as possible, often sympathizing and helping the weaker team. In undecided situations it could also start a smalltalk in order to motivate the players to speak and win its sympathy.

The sympathy of the CNPC for a team depends on the fulfillment of its wants. If it is dissatisfied, it may leave the team he was giving hints to.

The CNPC is supposed to memorize topics, preferred by users during the conversation. Later on, it would use this knowledge to ask questions about the
favorite topics of its preferred team.

4.3 Proposals for further development

4.3.1 The place of Ask and Answer in the comparison

The dialogue initiative in Ask and Answer should be mixed, with the CNPC asking quiz questions, but still allowing users to express their preferred topics. To achieve this, the CNPC should give players the chance to seize the initiative by not always directly asking a new question after an answer. It should be also able to understand and analyze user utterances that do not address the posed question.

The CNPC need not care about situations where some user expresses himself ambiguously. As designed, an adequate reaction in this case would be to simply play confused and bored and decrease the CNPC's affinity towards that user.

4.3.2 Improvements influenced by the discussed approaches

One of the core activities of the CNPC should be to give hints and to change sides. To achieve this, some type of user reflexion (model) is needed, as in HCA. Aside personal information (like name, gender and age), also representation of the preferred and disliked topics is needed. This could be a list of topic-preference pairs. So the CNPC could choose for example the topic that has biggest preference difference between both teams. Sympathy (or affinity) to each player could also enable some complex team sympathy behavior. For example, it may pay off to have one player in a team that says the things that are known to be interpreted negatively by the CNPC, but still need to be said, thus preserving the good image and relations of his teammates. The exact model for affinity calculation could be encapsulated as logical rules and be considered as the personal reasoning and emotional model of that CNPC, different models could formalize different approaches in human thinking. This could be used for example when a question is answered and closed. In this situation the CNPC should take the conversational initiative. Here are several example rules how it could decide what to say:

1. If the CNPC has a preferred team and knows what its favorite non-exhausted domain is, it could ask some non-posed questions in that domain.
2. if the CNPC has a preferred team, but does not know what domain it prefers, it could engage in a conversation on a general topic such as favorite book or movie. In this way, it motivates players to talk about their interests.

3. If the CNPC does not have a preferred team, it will start talking on its favorite topics (e.g. knowledge, games), so the teams could give opinions and thus he could chose sides.

<table>
<thead>
<tr>
<th>Façade</th>
<th>VICTEC</th>
<th>NICE Fairytale World</th>
<th>Mission Rehearsal Exercise</th>
<th>NICE H. C. Andersen</th>
<th>Ask and Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientedness approach</td>
<td>story</td>
<td>domain</td>
<td>story</td>
<td>task</td>
<td>domain</td>
</tr>
<tr>
<td>Dialogue initiative</td>
<td>system</td>
<td>system</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Number of speech acts</td>
<td>36</td>
<td>3 categories</td>
<td>8 ....</td>
<td>7+</td>
<td>N/A</td>
</tr>
<tr>
<td>Semantics extraction</td>
<td>topic (alternative to pragmatics)</td>
<td>integrated in pragmatics</td>
<td>N/A</td>
<td>Deep form-filling approach</td>
<td>topic, domain</td>
</tr>
<tr>
<td>Misunderstanding handling</td>
<td>concealed</td>
<td>N/A</td>
<td>N/A</td>
<td>repaired</td>
<td>ignored</td>
</tr>
<tr>
<td>User personality</td>
<td>artificial</td>
<td>N/A</td>
<td>artificial</td>
<td>N/A</td>
<td>HCA asks questions about the user</td>
</tr>
<tr>
<td>Emotional model</td>
<td>none</td>
<td>Ortony et al</td>
<td>none</td>
<td>EMA, based on Smith and Lazarus</td>
<td>own simplistic</td>
</tr>
<tr>
<td>Dialogue memory management</td>
<td>partially ordered non-replayable beats</td>
<td>plot milestones</td>
<td>information state</td>
<td>previous input and output turns; covered topics</td>
<td>covered topics</td>
</tr>
<tr>
<td>Gesture generation</td>
<td>hard-coded in content</td>
<td>physical act selection</td>
<td>N/A</td>
<td>deep character intention model</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3.1: The place of Ask and Answer among the systems under discussion

To determine the speech acts in use, one could start from some more general theories [Stockle et al 00]. The choice depends on the exact structure of the content. Acts like agree, disagree, not-certain, not-understand, thank, praise and oppose would be useful for sure. Like in the discussed systems, a statement
speech act without semantics would tell only that, for example, an answer was given, but this wouldn't help in deciding if it is wrong or right.

Thus the setting of Ask and Answer requires user input to be interpreted beyond its pragmatics. Minimally, a topic or keyword is required. This could either be done with pattern matching as in the case of dialogue acts or with part-of-speech tagging. However this could mean too great authorial effort. One alternative for making this process easier is to use ontologies (e.g. public ones - OpenCYC). For example when the user mentions the Bastille, the CNPC could not have any questions about it particularly, but could infer from the database that it is in Paris and played a role in the French Revolution, thus marking it as area of interest of the particular user. Once ontological queries (e.g. via OWL) are integrated in XEXPR, such inferences could represent another form of topic (category) reduction.

A theoretically approved cognitive emotion model is crucial for Ask and Answer. The game pretends to be socially educational and thus it needs to present a pragmatic and understandable behavioral model. This calls more for the use of rather shallow models that generate relatively adequate behavior. As part of this model, exceptional cases could be formulated too, in order to bring some needed effects in. However to keep things understandable, the model should describe direct dependencies in interaction and contain only a few exceptions. As a good place to start with this would be Orthony’s theory [Orthony et al 88], as it has influenced both theoretically-founded emotional models discussed here (VICTEC and MRE). I consider emotionality as the main tool to express personality. That is why, it is important to have it as independent as possible both from the technical system and the content domain. An example scenario could be that the CNPC gets happier when his preferable team wins, but fearsome for the end of the game if one of the teams gains too big advantage.

In Façade and NICE in order to handle repetitions, content is playable only once. This is also possible in the Storytelling platform. As in Façade, in situations where some content may need to be repeated, it is possible alternative formulations to be predefined. However in the Storytelling platform this involves combined visual editing of content states, manual editing of XML and use of predefined rule-sets, which is a confusing and error-prune process. Further development of the authorial tools and visual representation of content and
interdependence could made repetition handling better manageable. Once this is
done, other predefined rule-sets could imitate some partial order of states through
preconditions, similar to Façade. In the context of this parallel, with the
appropriate rule-set, one frame could be considered as an equivalent to a set of
beats with weights and priorities. Examples on how this could be managed are
already integrated in demonstrators of the Storytelling Platform.

The introduction of generic responses (like in VICTEC and MRE) would
also increase the expressive power and reduce authorial effort. Although some
exact (mainly informative) sentence formulations are inevitable (for example the
question “When did the French Revolution start?” is difficult for a machine to
reformulate), actions as greeting, affirmation, disappointment or appraisal could
be automated, as this is done in VICTEC in particular. For example in a situation
where the preffered team has answered wrong, in the story it could only be
defined that the CNPC should be dissapointed, and it would choose how to
express itself from a set of verbal and non-verbal behaviors, depending on its
emotional situation and game stand. It could just frown quietly or say “It was so
easy!” with a friendly smile.

4.3.3 Other proposals

As already discussed, the Storytelling Platform has a nested state
mechanism that needs further formalization in order to use the best of the
structure. Below are several levels of frame separation that would preserve the
story coherence. I present the levels top-down from highest level of abstraction to
exact question formulation:

1. Game frame – at this level one could define frames for introduction,
different game rounds, if any, and final awards. State update could be
managed by transitions at this level (Figure 4.3.1).
2. Domain and topic frames – probably several levels would be needed in the game rounds. Here different sets (frames) of questions in different domains and on different topics would be bundled. Several non-game frames (e.g. small talk, praise, insults, etc.) could also be placed at the upmost of these levels. Jess-based logic would easily allow complete interconnectedness of topics, optionally bringing connection sentences or gestures in-between. As an example one could consider the rules that take advantage the user model, discussed above.

3. Question frames – this would be a frame that represents a specific question. Here the author should place the question-posing and hint statements and reactions to right and wrong answers, as well as timeout. Optionally introductory background information or some other deviation could be placed also here.

Figure 4.3.1: Game frame level of the Ask and Answer transition network
A clear separation between state update and contextual information should be made in order to improve component capsulation. To achieve this, a single interface for external function definitions should be made. Then XEXPR and Jess could be adapted to use this interface.

So far the Stroytelling platform is deterministic, thus assuring the same scenario as a result to the same input. Probabilistic state selection would allow a more flexible plot, while still preserving the story coherence. It would also give the user a very effective feeling of unexpectedness. Such experiments have already been made in Façade and HCA.
5. **CONCLUSION**

The development of dialogue-driven storytelling systems is in its early stage. Although several complete solutions exist, they are rather limited to the authored content and difficult to be extended or migrated to other domains. Thus efforts are made to employ artificial intelligence and decrease the authorial effort. Special attention is paid to psychological theories as means to automate natural behavior in avatars.

In this work, I have described in detail and compared several dialogue management systems for interactive storytelling. I have made suggestions for the improvement and further development of the ZGDV Storytelling Platform and more specifically a game created with it - Ask and Answer. Although, because of different motivations and areas of application, a common comparison criterion does not exist, I have considered the main differences in the approaches and discussed their advantages in terms of applicability in Ask and Answer. I have put special effort in distinguishing and suggesting domain-independent and believably human sides of dialogue behavior.

I believe this works gives a good foundation for the further development of the ZGDV Storytelling Platform and particularly Ask and Answer, not completely solving all open problems, but addressing and giving proposals for the most important of them.
6. REFERENCES


6. [Crawford 04] Chris Crawford, 2004, Chris Crawford on Interactive Storytelling


10. [Gustafson, Lindberg and Lundeberg 99] The August spoken dialogue system

12. [Mateas and Stern 05] Michael Mateas and Andrew Stern, 2005, Structuring Content in the Façade Interactive Drama Architecture
18. [Iurgel and Ziedler 05] Ask&Answer: An Educational Game where it Pays to Endear your Capricious Virtual Companion


33. [Traum et al 03b] Toward Virtual Humans


35. [Traum, Marsella and Gratch 04] Emotion and Dialogue in the MRE Virtual Humans

