SYMBOLIC CONVERSATION MODELING USED AS ABSTRACT PART OF THE USER INTERFACE

Norbert Braun

Department of Digital Storytelling
Computer Graphics Center, Rundeturmstraße 6
64283 Darmstadt
Germany
Norbert.Braun@zgdv.de http://www.zgdv.de/distel

ABSTRACT

The art of conversation is a well-known interaction type between humans. Human-computer interfaces that follow this metaphor struggle with complex problems of speech understanding, speech generation and intelligent conversational behavior in general. This paper presents an approach that gives a simple, explicit symbolic model of conversation between human and computer to be used by interface designers as an abstract platform of conversational interaction – without being forced to regard the basic implementations of speech systems or graphical anthropomorphic avatars or virtual humans and therefore free from the problems of basic media manipulation.

Keywords: conversation, human-computer interaction, symbolic modeling, artificial intelligence, computer games.

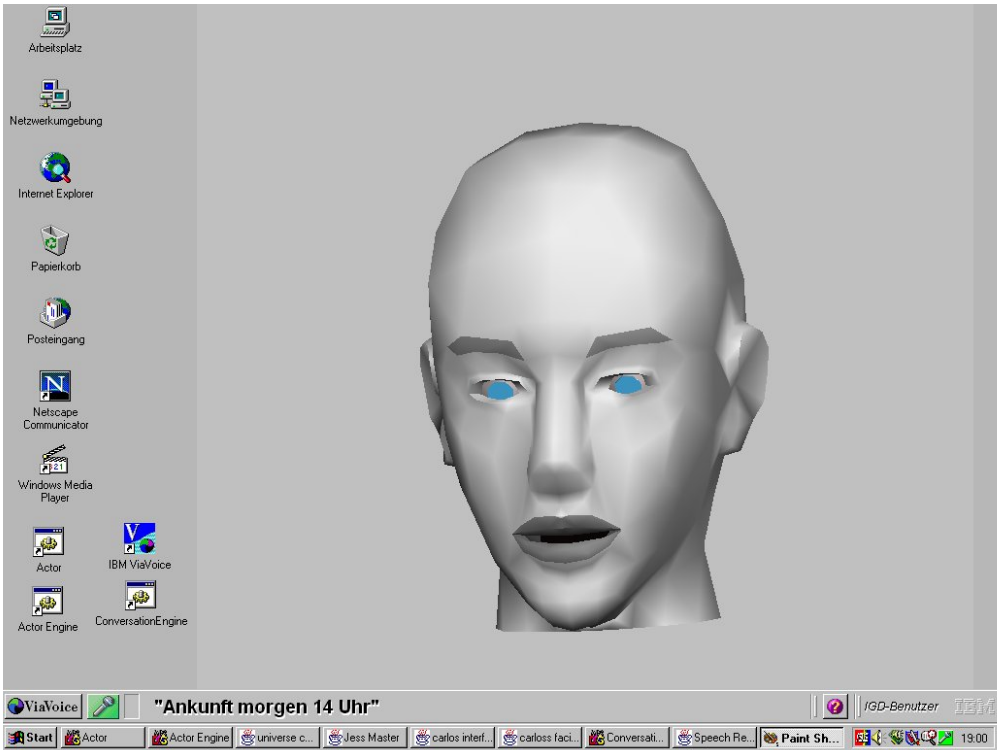
1. INTRODUCTION

Conversations are a well-known instrument of interaction between humans. Not surprisingly, conversational interaction between human and computer is a well-known metaphor of interaction in the computer science area – not only within research, but also within the imagination of ordinary people where movies have placed some sort of common agreement concerning how the conversation between human and computer should happen. Movies such as '2001: A Space Odyssey' (done in 1968 by Stanley Kubricks) with the (intelligent and dangerous) HAL computer or 'Star Trek' (produced by Gene Rodenberry, 1966-69) with similar advanced computers that talk to the crew - or that can be addressed by speech - affect the view of an advanced communication with the computer shared by generations. Unfortunately, these movies raise expectations of conversational interaction with a computer: computers have to understand natural language, they should be able to generate natural language, they should be friendly and educated with good manners, and they should be omniscient or at least very well oriented within their particular domain.

With these demands in the back of the people's minds, it is not surprising that researchers of the artificial intelligence (AI) domain who work with speech generation, speech understanding and domain knowledge modeling were the first people to generate conversational interaction approaches. Since computer graphics (CG) are becoming more advanced with virtual reality (VR) and 3D – modeling, resulting in human models that are nearly impossible to distinguish from real human (consider, for example, movies like 'Final Fantasy'), a primary demand on so-called virtual humans (or synthetic actors) is not only to look like real humans but to be able to communicate like real humans. Therefore, conversational interaction is also a research area within computer graphics. At least, many in the filed of psychology are becoming aware of the new possibilities created by AI and CG and are trying to transmit their knowledge about human-to-human interaction to the models of computer science. The psychological insights into human-to-human communication, however, have a symbolic characteristic. Researchers try to enhance their specific models of the AI or CG area with this data, but they fail in most cases. The reason for this failure is easy to understand: when applying symbolic data to low-level models of CG or AI, the data gets somewhat implicit – changes of the psychological models are very difficult to handle. To be both handy and applicable, the symbolic representations of conversational behavior should be explicit, not hidden within some AI or CG model.

Another problem of conversational user interfaces - and a fundamental problem of new interaction metaphors in general - is not the idea of the metaphor itself or the programming of its basic approach, but the ease of use of the metaphor when it swashes from the research area into industrial applications. Most research products are designed to be used by computer science experts who are, in addition, experts in the specific field of the metaphor. This is particularly true for the conversational user interface (CUI) metaphor. Interface designers have nice tools to style an effective WIMP user interface (WIMP – Windows, Icons, Menus, Pointing), but they have to do basic programming to make use of speech generation & understanding, control of virtual humans on the abstraction level of polygons or, if highly advantaged, on a task level (like wave hands or shake head). The level they should work on is something like: "Tell XYZ to the user." The virtual human should know his general conversational behavior and how to tell XYZ in the given context of the conversation.

The approach shown in this paper is an explicit symbolic model of conversation as a part of the user interface (UI). The approach can be used in advance as a conversation engine (CE): the application programmer or user interface designer can simply tell the CE which contents to present – the CE will manage the conversational characteristics of the human-computer interaction in real time. Of course, this demands some higher intelligence of the input and output modules of the user interface. In this paper, the CE is placed as a separate module in the context of a CUI that consists of virtual human engines, combined with speech generation applications, as well as user interpreters combined with speech understanding modules.



"Ankunft morgen 14 Uhr"

Figure 1: This shows a prototype implementation of the MAP user interface agent. Speech understanding and speech generation - the language is German, but can be effortlessly replaced with English - are used in combination with a synthetic actor. The components are driven by the conversation engine.

The following paragraphs show applications that use the conversational metaphor, discuss the approach shown within this paper, describe its implementation, show how the CE is working together with input and output modules of the UI (giving an architectural overview) and give a conclusion with some words on future work in this area.

2. APPLICATIONS AND RELATED WORK

Computer graphics research results in systems like Crawford's Erasmatazz [Craw00] that implement conversation on the basis of actor behavior. Unfortunately, the system is not handy for authoring conversations. Cassell's REA [Casse99] is an approach on the basis of discourse modeling. Cassell is using rule-based generation within REA for numerous conversational aspects of agent communication with a strong relationship to the possibilities and goals of the agents. Therefore the conversations are story-related, but the behavior is preprocessed - there is no real time generation.

A specialized conversational approach is shown by the DIVA II project [Braun00]. There, the conversation takes place within a video presentation as the conversational limitation factor of the system. The approach is based on audio and video annotations via hyperlinks – so-called video hyperlinks and audio hyperlinks [Braun99]. The application shows that the conversational approach can be handled completely without speech input. Within the DIVA II project, the conversation is modeled implicitly within the video and audio data annotations.

Conversations between several synthetic actors (some virtual, some physical) and a user is shown with ZGDV [Spierl99] inquiry kiosk / trade show kiosk. The conversation is not limited to speech only; several modalities with unorthodox input devices (i.e., a physical book, among others) are used. The conversation is modeled implicitly in 3D VRML (Virtual Reality Modeling Language) data; therefore, any changes can be very expensive.

In industry/commerce, there are applications like EMBASSI (Multimodal Assistance for Infotainment and Service Infrastructures) which use an extended speech-recognition/generation controller (so-called dialogue engine or 'DE' based on AI research) [Ludwig01] as the basis of conversational interaction [Alexa00]. Within EMBASSI, the conversational behavior of actors/virtual humans is implicitly modeled as a part of the DE.

The project MAP [Gerf00] (a basic platform for agent technology as the approach to the multi-media workplace of the future) is a combination of research and industrial development. It implements the conversation engine described in this paper as a part of its user interface agent (UIA), like all other components shown in this paper - see figure 1. The conversation module within the UIA models conversation primarily on an abstract symbolic level, completely independent of the virtual actors' possibilities and goals. This allows the separation of the authoring of stories, the separation of the goals and possibilities of agents, and the adjustment of the system's complexity to a handy grade.

3. ABSTRACT CONVERSATION MODEL

Conversations depend on diverse factors. These factors are directly deduced from the behavior within human-to-human communication. To deduce these factors, we have analyzed numerous videos, pictures, and books about the psychological and social aspects of conversation. For example lists of intuitions of conversation participants are derivate from video, analyzed for their visual effects, transferred by designers to a first set of behaviors; these to be the basics of the animation of conversation behaviour. We even analyzed books like [Molcho01] to get a description of the non-speech behaviors of humans. The factors are listed (but not restricted) in the following points:

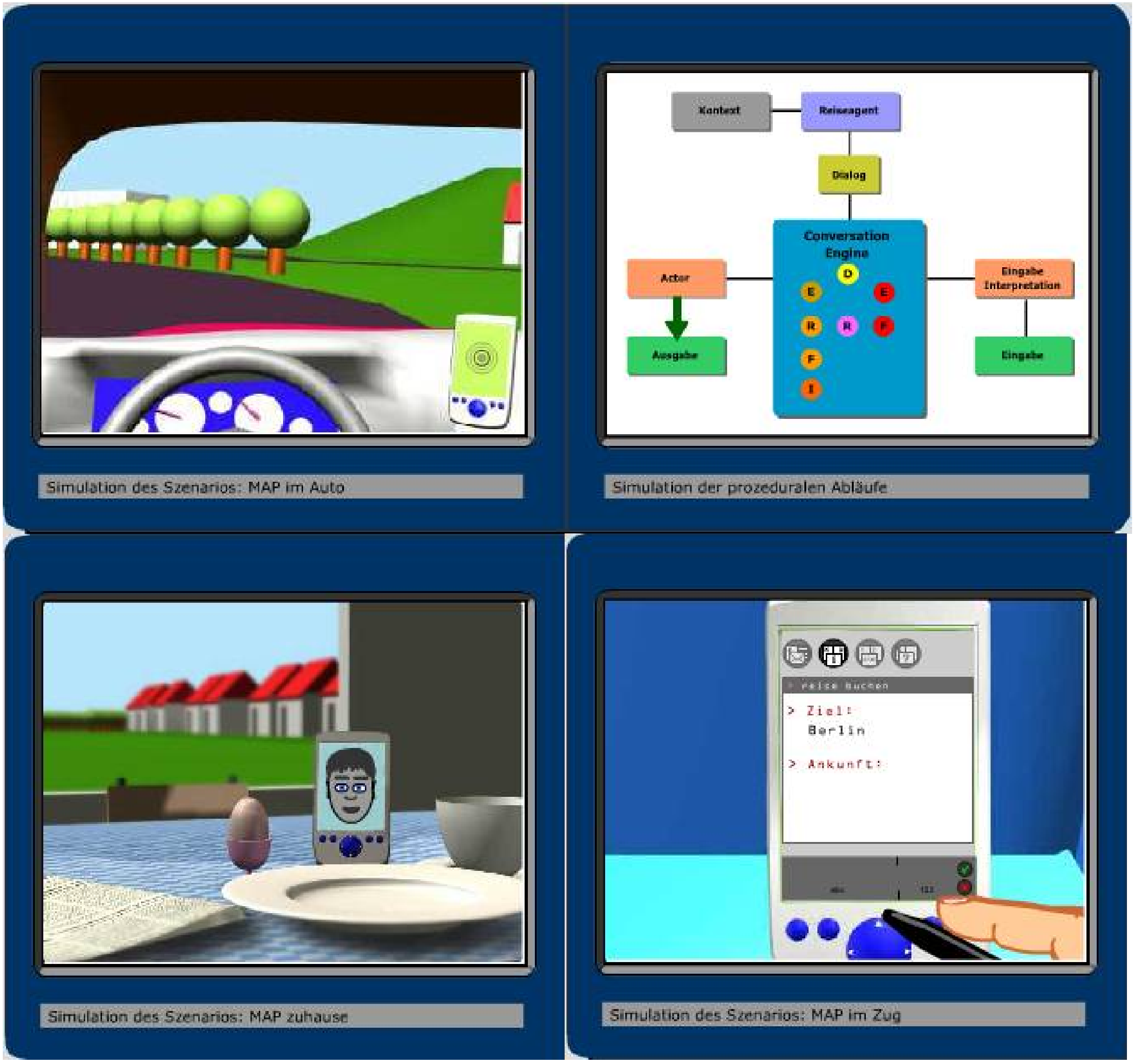
- social and emotional aspects: like ranking, relationship.
- story and immersion: sequences to be told or question-answering, disturbance possibility of interactive movies related to the case of a virtual assistant.
- the actual focus of the conversation participants (CP): does the CP look at the virtual actor or is he looking towards the front windshield while driving the car?
- content-related aspects: is the actual content within a conversation discourse a question, an answer or some simple statement; does it have some relation to other content in the past or the future of the conversation?
- navigational aspects: opening or closing of a conversational discourse, turn taking, getting attention.

It is obvious that the factors are very abstract and symbolic; it seems that the content knowledge is minimized while the knowledge of conversational discourses and user behavior is maximized.

A notable aspect is that the conversational aspects are described without regard to the modality or the medial expression of the content to be presented. The media problematic appears on a lower application level – at the various presentation modules for media like video and audio-visual presentation of synthetic actors. Of course, the problematic has to be solved within these levels – and it is solved within the MAP project; see figure 2. By hiding the media problematic, the CE is very easy to use, even for non-experts in the conversation domain.

Conversation modeling is somewhat orthogonal to content generation – within the CE, how to present the content is defined; the content generators and managers define what kind of content is to be presented. Therefore, there is a strong separation between content generation and communication process. Especially the CE is not modeling the KQML [Labr94] extensible set of performatives. (These peerformatives define the permissible operations that agents may attempt on each other’s knowledge and goal stores. The performatives comprise a substrate on which to develop higher-level models of inter-agent interaction, such as contract nets and negotiation.) The CE is modeling a special part of the behavioral ‘thinking’ of the computer.

Conversation modeling is divided in the description of a specific conversational situation and the transfer of one conversational situation to another desirable conversational situation. Both parts together describe a conversational discourse.



Simulation des Szenarios: MAP im Auto

Simulation der prozeduralen Abläufe

Simulation des Szenarios: MAP zuhause

Simulation des Szenarios: MAP im Zug

Figure 2: The picture shows how amodal input of the conversation engine (upper right) is used to produce conversational behavior on a speech-based (upper left, scenario within a car), a synthetic actor-based (lower left, scenario at home) and a GUI-based (lower right, scenario within a train) output engine. The scenarios are part of the MAP project.

The conversational situations are described with the following aspects (these aspects could be named ‘conversational acts’, but there is no close relation to so-called speech acts) – every aspect is shown with its name, attributes, and a short description.

- Conversation participant

name
type ∈ {active, passive}
turn ∈ {true, false}

This fact is describing a participant of the conversation. This could be the user or an automated actor independent of its medial characteristics.

- Conversation element:
  behavior ∈ { Open, Talk, Listen, PutTurn, GetTurn, StartSequence, EndSequence, GetAttention, ChangeDiscourse, Close},
  sender,
  recipient list,
  content,
  discourse,
  timeslot,
  intensity ∈ {force, neutral, smooth}

  This simply means: Someone (the sender) is doing something (the behavior and the content) to someone else (the recipient-list) at a specific point of time within a specific conversational discourse.

- Content:
  name,
  type ∈ {Question, Answer, Term}, reference,
  status ∈ {todo, do, stopped, done}, discourse,
  priority ∈ {low, medium, high},
  importance∈ {low, medium, high}

  The content is of the form answer (related to some former conversation element), question (related to some future conversation element) or term (related to the moment).

- Answer extends Content:
  repeat allowed ∈ {true, false}

- Question extends Content:
  repeats

  Repeats show the number of times the question was given to a user.

- Story
  name
  type ∈ {sequence, asynchronous}
  content-list,
  status

  This is a set of content elements related to each other in some way. For the conversation the type of relation or what the content describes is not important – for the conversation, only the linearity or non-linearity of the presentation is relevant as non-linear content presentation is done by request (user asks for the information) and linear presentation is done automatically (content is shown to the user as long as the user does not interrupt).

- ThinkAbout:
  reference

  This means that a specific content (the reference) causes a problem that can not be solved by the conversation engine, e.g. there is a user question without an actual answer. Then some story module is informed via that fact.

- Time:
  timeslot list,
  actual time,
  new time

  As the conversation engine is driven by beats, every beat has its own symbolic point in time (actual time). Behavior created within a run is timed to a new time point (new time)

- Discourse
  name,
  timeslot

  Of course several discourses can be held by one conversation engine. Every discourse is described by its name and start time.

Complex conversational scenarios can be constructed with those relatively simple facts – scenarios for every kind of synthetic actor, even user behavior can be described with those facts (and simulated with the conversation engine). To transform one scenario into another – e.g. to transform a scenario where the user is asking a question, to a scenario where the system demands the turn, to a scenario where the user is giving the turn, to a scenario where the system is answering the question – a large set of rules is developed. There are rules for many conversational situations like the linear/nonlinear telling of content, question/answer situations, opening/closing a conversational discourse, jumping from one discourse to another, turn-taking-behavior.

Rules can be applied to the conversation engine very easily as rules are defined for special situations – so far, the conversation system is easy to expand by simply adding specified rules. One of the rules is shown (simplified) in the following:

```
(Rule Statement_with_reference

    (ConversationParticipant: It is my turn and I have the
    focus)

    ?fact_content_statement<-
    (content: There is a content with a reference to a
    content given by the user)

    (ConversationElement: The user sent the content to
    me)

    (ConversationElement: I'd opened the discourse)

=>

    (assert (ConversationElement: Present the content) )

    (modify ?fact_content_statement: Status of content is
    do)
)
```

This way, the system is simple to understand for a conversation designer: The left side of a rule matches a conversational situation (a conversational aspect is named; its properties to be matched against the conversational situation are listed); the right side of a rule gives the modifications of the conversational situation; that means the rule is a function with the conversational situation as argument, with its result as another conversational situation. In order to keep this easy way of understanding, the system is designed in the style of a traditional knowledge base system: the conversational situation is stored as a set of facts; the conversation processing is stored in associated rules.

4. ARCHITECTURE & IMPLEMENTATION

The system discussed in this paper is embedded within a storytelling system (with the MAP UIA as a special instance of the system) that balances content-related (also called story-related) and conversational aspects:

- authoring control of the story and conversational situation
- automated storytelling
- automated conversation modeling
- processing of user input
- actor control
- media management

The AI of the system is modularly distributed onto three layers – the story (or content-giving application) as a strategic level, the conversation as the operational level, and the user input interpreters and actor engines at the executing level.

Within this environment (as shown in figure 3), the conversation is a function of application-specific content and user input. Content and user input are processed by separate units and are given continuously as abstract input to the conversation engine. The conversation itself controls the actor response, as well as the media presentation of the system. The conversational output is mapped on the actors' possibilities as a final step: thus far, the conversation modeling is nearly independent of the actors' possibilities. The actors process their conversational input within an actor engine. This simplifies every conversation layer and makes it handy for a conversation designer - a concept suggested by the game industry, see [Wood01].

As previously indicated, we specialize our point of view of conversations to the content-independent behavior of the conversation participants. As the independent behavior can be seen as special knowledge about conversational situations, we organize a conversation as a knowledge transformation problem.

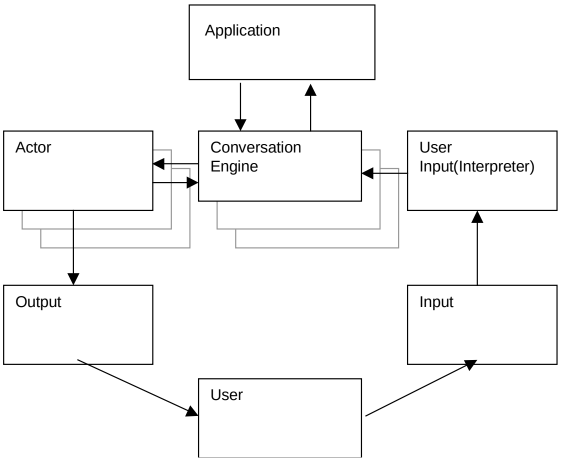


Figure 3: Architecture of the User Interface Agent (based on the project MAP)

This fits well into the actual trend of rule-based finite state machines within game design, see [Wood01]. With a couple of predefined rules, relatively complex behavior can be generated – without path planning problems and with simple debugging.

The conversational knowledge is represented by rules; the conversational situation between user and system is represented by so-called facts. (A fact is a symbolic description of a part of the knowledge base; the fact properties are stored in so-called slots. A rule is a kind of if-then construct, the if-part matches to the knowledge base, the then-part represents a knowledge base modification). As knowledge base, we use the Jess (Java Expert System Shell) Engine [Fried01].

Of course, the application programmer is not interested in details of the conversation generation. Therefore, the general task of the application programmer, using the CE, is to give the content to be presented by the CE. The CE has an API (application program interface) with a (simple-to-use) functionality to generate a discourse with its conversation participants, as well as to annotate the content with (simple) meta-data, such as affiliation to a story or characteristic of question or answer. The following XML syntax-styled documentation shows how the API is working (strings are marked with a $-sign):

```
<Discourse name="$DiscoursName$">
        <conversationparticipant>$Name$
</Diskurs>

<Story name="$NameStory$" typ="$TypStory$">
        <content> $content$ </content>
```

```
</Story>

<Content name="$content$"  typ="$TypContent$"
ref="$refContent$" discourse="$DiscoursName$"
priority="$ThePriority$"
importance="$TheImportance$">
        <what> $here the content...$ </what>
</Content>
```

Of course there is an extension of the Content-Tags to define specialized content like question (identical to content) and answer:

```
<Question name="$content$"  typ="$TypContent$"
ref="$refContent$" discourse="$DiscoursName$"
priority="$ThePriority$"
importance="$TheImportance$">
        <what> $here the content...$ </what>
</Question>

<Answer name="$content$"  typ="$TypContent$"
ref="$refContent$" discourse="$DiscoursName$"
priority="$ThePriority$"
importance="$TheImportance$" repeat-
allowed="$True/False$">
        <what> $here the content...$ </what>
</Answer>
```

With that API, the application programmer can shift his content to the CE without regard for how to present the content in a conversational way, see figure 4.

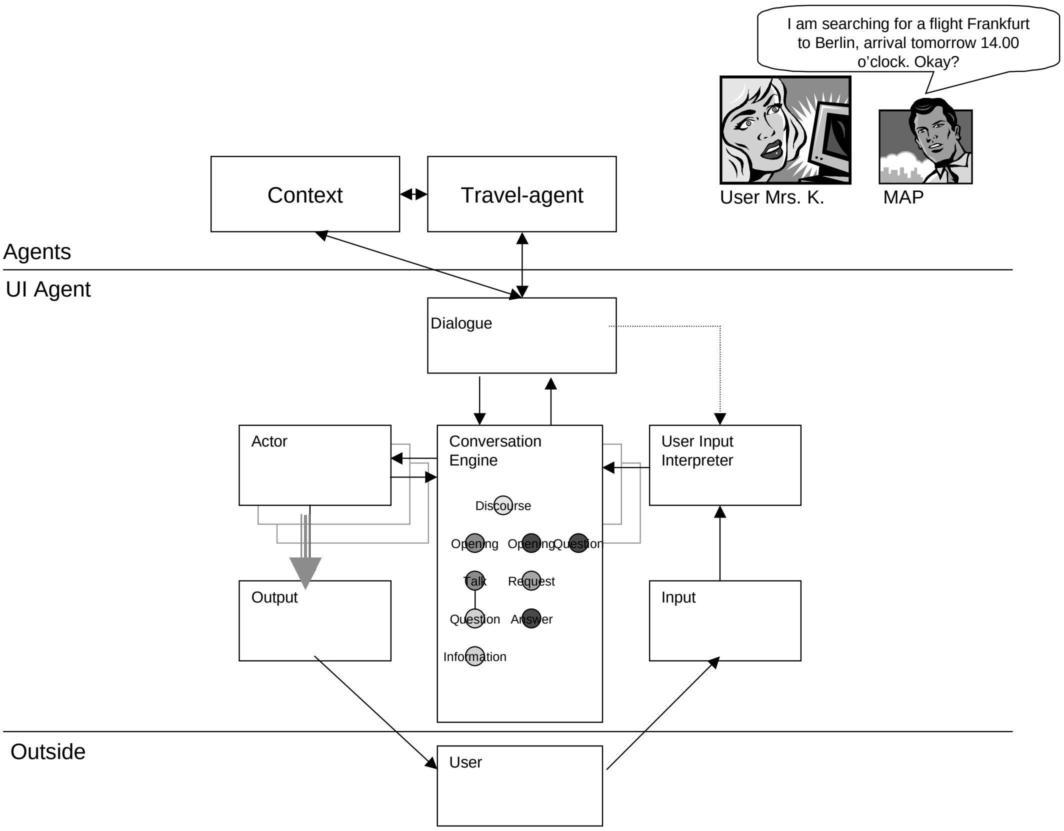


Figure 4: (simplified) Example of conversational aspects handled by the conversation engine within the MAP project. The agents of the MAP system are using the UI Agent as their general conversational UI

The MAP User Interface Agent was tested in regard to its effectiveness, efficiency and acceptance to the user by Siemens Usability Lab [Sand01] with very optimistic results. The analysis showed that the users accepted the MAP UIA as a personal assistant. Users find it helpful to get access to the MAP System via multi modal conversation and used multi modal inputs to communicate their objectives in a subjective and objective efficient way. In general that test shows that the conversation metaphor can be profitable used for delegation and assistance.

5. CONCLUSION

Within the paper, a basic approach for the general use of the conversational metaphor is shown. The approach is based on the processing of symbolic conversational data to allow easy access and modification of conversation rules within the conversation model by the conversation designer. This is a very important fact: as user-machine conversations are in an early stage of research by computer science, but in a very late stage in the so-called humanities, the likelihood of changes in several conversational aspects is very high. The conversation model stands and falls with its ease of maintenance.

The approach is implemented within a storytelling architecture. This shows that the conversation engine offers an API to the application programmer for easy integration into applications as well as an interface to user input interpreters and general output presenters like virtual human engines. In advance, it shows that it generally works as the amodal part of the multimodal system.

This approach is used in commercial projects like the MAP project (industrial/commercial application). Future work will be done by modeling specialized conversational situations, therefore by increasing the number of the conversational situations that can be handled by the conversation engine.

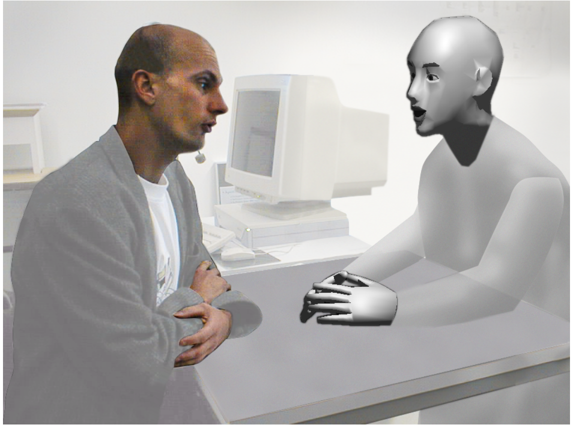


Figure 5: Vision of Conversational Human-Computer Interaction

Finally, one thing is obvious – the development of conversation engines that can be easily accessed by content providers is especially useful for those parts of the UI that can not be handled by 'traditional' window-styled UIs – these are the delegation and assistance tasks that are difficult to define via a window, but easy to define in a more general way via conversation. This leads to the claim of highly user adaptable, highly flexible content-providing applications that work in close relation to the conversation modeling; this is to give the user the intelligent, eloquent conversation participant that he knows from TV and the movies, see figure 5.

REFERENCES

[Alexa00] Alexa, M., Müller, W., Spierling, U., and T. Rieger: Face-to-Face With your Assistant. Realization Issues of Animated User Interface Agents for Home Appliances, *Conference on Intelligent Interactive Assistance & Mobile Multimedia Computing*, Germany, 2000

[Braun00] Braun, N.: Interaction Approach for Digital Video Based Storytelling, *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Czech Republic, 2001

[Braun99] Braun, N., and R. Dörner: Temporal Hypermedia for Multimedia Applications in the World Wide Web, *3rd International Conference on Computational Intelligence and Multimedia Applications*, ICCIMA ´99, New Delhi, India, 1999

[Cassel99] Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjalmsson, H., and H. Yan: Embodiment in Conversational Interfaces; REA. In *Proceedings of the CHI '99*, pages 520-527, ACM Press, Adison – Wesley, USA, 1999.

[Craw00] C. Crawford: Understanding Interactivity, *http://www.erasmatazz.com/*, 2000.

[Fried01] Friedman-Hill, E. J.: Jess, The Java Expert System Shell, *SAND98-8206* (revised), Sandia National Laboratories, Livermore, CA, USA, 2001.

[Gerf00] Gerfelder, N.: MAP: Multimedia Workspace of the Future, *Computer Graphic topics*, 3/2000, pp 10-11, Germany, 2000.

[Labr94] Labrou, Y., and T. Finin: A semantics approach for KQML -- a general purpose communication language for software agents, *Proceedings of theThird International Conference on Information and Knowledge Management* (CIKM'94), November 1994

[Ludwig01] Bücher, K., Forkl, Y., Görz, G., Klarner, M. and B. Ludwig: Discourse and Application Modeling for Dialogue Systems, in: G. Görz, V. Haarslev, C. Lutz, R. Möller (eds.): ADL-2001. *Proceedings of the KI-2001 Workshop on Applications of Description* Logics, Vienna, Aachen 2001.

[Mateas97] Mateas, M.: An Oz-centric review of interactive drama and believable agents, *Technical Report*, School of Computer Science, Carnegie Melon University, 1997.

[Molcho01] Molcho, Samy: Körpersprache im Beruf, *ISBN 3442163269, Goldmann*, München 2001.

[Sand01] Sandweg, N. and D. Hermann: Analyse der technischen Konzeption: Usability Anforderungen, ZE 5.3.1, MAP Konsortium, 2001.

[Spierl99] Behr, J. and U. Spierling: Conversational Integration of Multimedia and Multimodal Interaction, Es*say Computer Graphik Topics*, Darmstadt, Nr. 4/1999.

[Wood01] Woodcock, S.: Game AI: The State of the Industry, *Game Developer Magazine*, USA, August 2001.