Design and First Tests of a Chatter

Hans Dybkjær[1] and Laila Dybkjær[2]

[1]Prolog Development Center A/S (PDC), H. J. Holst Vej 3C-5C, 2605 Brøndby, Denmark
dybkjaer@pdc.dk
[2]Nislab, University of Southern Denmark (SDU), Campusvej 55, 5230 Odense M, Denmark
laila@nis.sdu.dk

Abstract. This paper presents the design of an over-the-phone, non-task-oriented, conversational spoken dialogue system character created to chat with users about himself and entertain them. Focus is on techniques used in personality modelling of the system character, how his mood changes based on input, and how this is reflected in his output. Experience with a first version of the system and planned improvements are also discussed.

1 Introduction and background

During the construction of a telephone-based, task-oriented frequently asked questions (FAQ) system concerning holiday allowance [7], we discussed the issue of how to recognise user dissatisfaction from what users say and possible reactions from the system to such input. In this context we felt a need to experiment freely with the limits and possibilities of the technology available to us which was a standard SpeechMania platform with a Danish language package.

The platform comes with a speech recogniser, a parser, support for recording output phrases, a Speech API for plugging in a speech synthesiser, and the grammar and dialogue description language HDDL. An application is developed by writing a dialogue model and a set of grammars in this language. The dialogue model and the grammar determine the application domain. To model emotions we could exploit features like (phonetic) lexicon, grammar, phrasing, recognition scores and dialogue flow, but the platform does not support exploitation of e.g. glottal stop, stress, prosody or non-linguistic vocal phenomena like laughter or mood.

A result of the experiments was a non-task-oriented conversational spoken dialogue system (SDS) whose character, soon named Crazy-Kurt, is childish, affective, self-centred, and defensive with an underlying uncertainty. His mood and temper is likely to vary considerably during a dialogue. The main goal of the Crazy-Kurt system is to entertain users. The user may call Kurt and chat with him in Danish, primarily about his preferences of food, notably fruit and vegetables. Kurt also knows a few other things about himself, e.g. his name, his age, and where he works.

Crazy-Kurt was running on a demo-line at PDC for quite some time. He was called by colleagues, children of colleagues, and sometimes by customers during dinner parties. To customers Kurt was described as follows:

> "You may call Kurt and get acquainted with him. Depending on your input he changes his attitude and becomes cross, embarrassed, or happy. If he perceives your input as negative, he becomes cross, if you praise him he becomes happy, and if you ask embarrassing questions he becomes embarrassed. The system uses a limited vocabulary (350 words, but it may without problems be increased to a couple of thousands words) and a mixture of speech synthesis and pre-recorded speech. The speech synthesis is used to tell the caller what the system recognised, which can lead to entertaining situations."

We had much fun working with this system. Kurt was inspired by users' sometimes quite rude input and weird behaviour. Contrary to typical task-oriented systems Kurt is allowed to behave in an impatient, selfish and impolite way, i.e. he has the same rights as users to express his emotions and opinions about the interlocutor.

A FAQ system is in may ways very different from the Crazy-Kurt system. For instance, it must be polite and patient to users and appear serious and professional. Nevertheless, the experience we got from experimenting with the Kurt system both concerning technology and design created useful input to the FAQ system, e.g. regarding adequate reactions to the emotions users express.

There is still much work which could be done on the Crazy-Kurt system and many possible improvements. However, we find it worthwhile to report on our work now since we believe that many of the ideas behind Crazy-Kurt are of broader interest and have so far received little attention in the area of SDSs.

We describe the design of and first experience with Crazy-Kurt. In particular we discuss techniques used in the modelling of his personality, including how input affects his mood and how his reactions are reflected in his output (Section 2). We discuss lessons learned from experiments with the system (Section 3). Section 4 briefly presents related work. Section 5 concludes the paper and discusses future work.

2 Personality modelling

With Crazy-Kurt we wanted to create an entertaining, chat-like (non-task oriented) conversational character. The entertainment value was supposed to come from the dialogue and the way it evolves, and we wanted this value to be apparent from early on in conversations with Kurt. With conversational dialogue follows an increased risk for low recognition scores. However, both entertainment value and low recognition scores may be handled via childish and affective characteristics as follows:

- Self-centred: He only wants to talk about himself and his taste for fruit and vegetables. This has the advantage that it helps delimit the domain.
- Affective and impatient: His mood is quite easily affected, which makes it less visible that the system not always understands what the user says.
- Direct: Kurt may be rude and tactless in the way teenagers often find entertaining.
- Embarrassed: Some topics are perceived as embarrassing by Kurt. It is irrational, but he reacts by becoming annoyed and by changing topic in such situations.
- Defensive and evasive: Towards the interlocutor Kurt behaves in a confident and self-assured way. However, internally he is very uncertain. He will never admit mistakes. If Kurt does not understand the input, he will jokingly change topic.

The characteristics listed above are taken into account in the definition of dialogue structure and output. The personality modelling uses the following means:

1. The *dialogue flow* is based on dialogue acts, input parameters, and affective state.
2. Affective state is modelled via *satisfaction*. Many other emotion parameters exist, e.g. hunger, anger, happiness, tiredness, and patience, but we wanted to start the experiments with one main parameter.
3. Self-confidence, or just *confidence*, relates to recognition score values and user confirmations. High scores and user acceptance imply high confidence.
4. *Input parameters*: The lexicon, more precisely nouns and adjectives, are emotionally modelled in terms of face value, preference, and embarrassment. The lexicon (and grammar) also determines what Kurt can understand.
5. *Satisfaction changes* are computed from confidence and the input parameters. They are a key parameter in exposing affect and mood changes.
6. *Output phrases* are designed in accordance with the personality, including rudeness, self-centredness and jokes. Phrases are varied according to the satisfaction and satisfaction changes. Feedback quotes user input directly using synthetic speech which also may be entertaining.

Each of these modelling means is explained in more detail in the following sections.

2.1 Dialogue flow architecture

Kurt's conversation is structured in terms of a simple state-model, cf. Figure 1. Depending on the contents of the user's input and the current satisfaction state, the Kurt system will select a path through the dialogue model and output utterances expressing the dialogue acts noted in each of the output boxes it passes on its way, see also Section 2.6.

2.2 Satisfaction

The central part of Crazy-Kurt's personality model is his satisfaction level which varies on a scale between –5 (angry) and 5 (exalted). Regarding Kurt's reactions the scale is divided into discrete intervals. At both ends the emotions may overflow. At the negative end Kurt will hang up while at the positive end Kurt will get the impression that the user is making a fool of him and will suddenly change to an angry mood. This is done to simulate a personality that is affective and seemingly self-confident but internally uncertain.

The initial state is a neutral satisfaction level where Kurt is neither particularly angry nor happy.

2.3 Self-confidence

Kurt's self-confidence affects his reaction patterns and the magnitude of his satisfaction level changes. His self-confidence expresses how confident he is in having interpreted the user's input correctly.

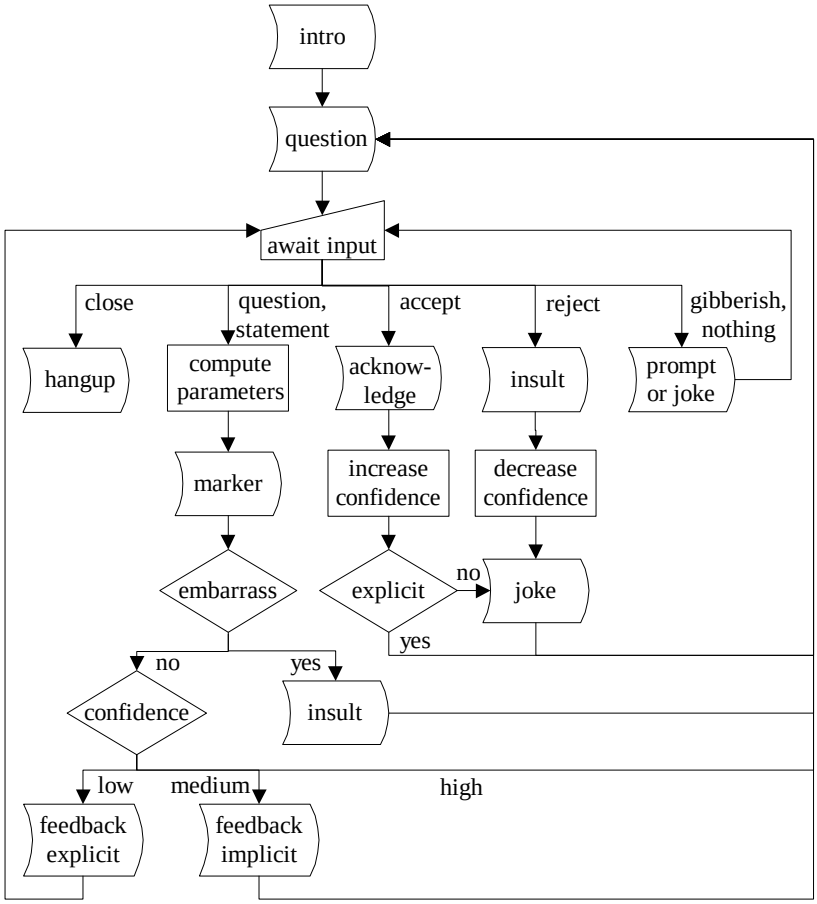

Figure 1. Flow of the dialogue. The test "explicit" checks if the previous system turn ended with explicit feedback.
Legend: prompt output, action, input, condition, condition transition flow (with test).

Confidence maps into the recognition score scale (Figure 2) and determines the type of feedback: For each input turn the recogniser provides a recognition score between 0 and 1. A low recognition score will imply explicit feedback to what the user said, and a low confidence will result in explicit feedback to a larger interval of low recognition scores.

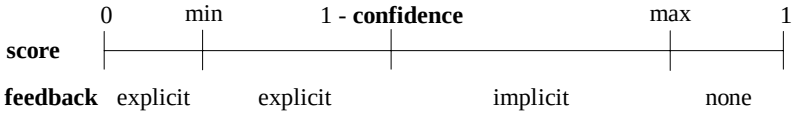

Figure 2. The feedback type is a function of confidence and recognition score.

The level of self-confidence is modified depending on the smoothness of the dialogue. In the current implementation only a fairly simple version of smoothness is realised, modelled via user reactions to explicit and implicit feedback. If the user agrees, Kurt's confidence grows, while disagreement makes his confidence decrease. Reactions to explicit feedback has a larger effect than reactions to implicit feedback. The present model does not include other smoothness features in terms of topics and the flow of dialogue such as re-phrasings, unexpected input acts, or loops.

2.4 Input parameters

Kurt's degree of satisfaction continuously changes not only depending on self-confidence but also depending on the current input topic and on the course of the dialogue both of which influence the input parameters used.

Every word in the system's lexicon is tagged with a face value [-5; 5], preference [-5; 5], and embarrassment (boolean). Below we explain how face value, preference and embarrassment are calculated for an entire user turn. User input is assumed to be a concatenation of word sequences covering one or more concepts. Values are calculated separately for each concept. See Table 4 for an example of a dialogue annotated with input parameter values.

In addition we discuss the implications of negations in input which affect both face value and preference.

2.4.1 Face value

Kurt is very sensitive to losing face. Certain kinds of interaction from the user have a negative face value, e.g. corrections and insults. Other kinds of interaction have a positive face value, e.g. praise will boost his self-confidence. The full analysis is complex, and we have used a simplification by letting face value be a function of the input word face values: If any input word face value is negative, take the minimum. Otherwise, take the maximum.

2.4.2 Preference

Certain words/topics have been defined to denote something that Kurt does not like. Other words/topics are neutral, while others again denote something Kurt really likes.

For input with a neutral face value, preference is independent of the face value. For input with other face values there will be a knock-on effect so that if the face value is positive, input denotes something Kurt likes, and vice versa for negative face values.

Preference cannot be calculated as a function of single words. For instance, a preference for red hair is not a function of the preference for red and the preference for hair. All verbs, adjectives and nouns have attached a preference but combinations need a separate assignment of preference. However, we have used the same simplification as for face values and compute preference as a function of the input word preferences: If negative preferences exist, take the minimum, otherwise take the maximum.

2.4.3 Embarrassment

Certain words/topics have been defined as embarrassing to Kurt. All other words/topics are neutral in this respect. In case of embarrassing input Kurt will try to talk about something else.

If any input word is marked as embarrassing, the entire input is perceived as embarrassing by Kurt.

2.4.4 Negation

Negation changes face value and preference. It makes a considerable difference whether the user says "you are stupid" or "you are not stupid". But it does not affect embarrassment since the embarrassing topic is mentioned no matter if negated or not.

There are two types of negation. One type is syntactic such as constructions with "not", e.g. "you are not clever". A second type is semantic, realised via the use of verbs that are "negative", e.g. "do you hate oranges". The implication of a negation depends on whether the utterance is a statement or a question, cf. Table 1.

Table 1. Implications of negation in statements. The effects of questions without a "not" are identical to those of the corresponding statements and are omitted.
*) Note that in Danish "elsker du ikke frugt" can only be a question, not an imperative.

User utterance	**Concept**	**Preference**	**Negative verb**	**Not**	**Change**
Statements					
you love fruit	fruit	positive	no	no	positive
you don't love fruit	fruit	positive	no	yes	negative
you hate fruit	fruit	positive	yes	no	negative
you don't hate fruit	fruit	positive	yes	yes	positive
you are stupid	stupid	negative	no	no	negative
you are not stupid	stupid	negative	no	yes	positive
you hate ice	ice	negative	yes	no	positive
you don't hate ice	ice	negative	yes	yes	negative
you like shoes	shoes	neutral	no	no	neutral
you don't like shoes	shoes	neutral	no	yes	neutral
you hate shoes	shoes	neutral	yes	no	neutral
you don't hate shoes	shoes	neutral	yes	yes	neutral
Questions with "not"					
don't you love fruit*	fruit	positive	no	yes	positive
don't you hate fruit	fruit	positive	yes	yes	negative
aren't you stupid	stupid	negative	no	yes	negative
don't you hate ice	ice	negative	yes	yes	positive
don't you like shoes	shoes	neutral	no	yes	neutral
don't you hate shoes	shoes	neutral	yes	yes	neutral

2.5 Satisfaction changes

The input parameters face value and preference will change the level of satisfaction into a positive or negative direction, with an adjustment based on self-confidence. The third input parameter, embarrassment, is only used to side-step, cf. Figure 1. The calculation of the change to satisfaction per turn is made as follows:

```
contribution = (face value + 0.5 * preference)/1.5
change = (score >  1-confidence)
       ? (1.5 * contribution)
       : (0.8 * contribution)
```

The weights used may of course be adjusted. Those presented here give reasonable results in the Crazy-Kurt system.

2.6 Output phrases

The output is based on canned phrases. They need to be tailored to the system personality and to the target user group. For example, a system which targets teenage users may use expressions commonly used by teenagers. The generation of an output turn includes the following rules:

- The output is a concatenation of utterances depending on the user's input and in accordance with the flow expressed in Figure 1. See also the dialogue in Table 4.
- Each utterance includes a dialogue act which is either question, prompt, joke, insult, feedback, or acknowledgement.
- The style level of each utterance depends on the current satisfaction level.
- For each act type and style level, the concrete phrase is selected randomly from a predefined set of phrases (typically three).
- In the feedback, the user input phrases are quoted using synthetic voice, all other output is generated as pre-recorded phrases.

2.6.1 Change markers

The most elaborate phrase generation is that of change markers which are small expressions or exclamations that mark a change in the level of satisfaction towards the user. These markers vary in accordance with the change in satisfaction (Table 2). See also the examples in the dialogue in Table 4. A couple of notes:

- There is a clear difference between the suspicious markers at (angry level, exalted change) and the tolerant markers at (exalted level, angry change).
- However, the phrasings can only be approximate, and the same phrase may occur in several states.

Table 2. Examples of change markers.'-' means no marker.

Change / **Current level**	Angry	Cross	Neutral	Happy	Exalted
Angry	You moron.	Listen you fool.	-	Hrmf.	Why should I believe you?
Cross	Listen, you fool.	Bah! Stupid!	-	Hm, perhaps.	That's better.
Neutral	Bah! Stupid!	-	Well	-	Okay!
Happy	Hello there.	Stop it.	-	Fine, stay on.	Great!
Exalted	Now, now.	What now?	-	Great!	You are my best friend!

3 Lessons learned from collected dialogues

As our focus has been on design experimentation and technology exploitation, no controlled experiments have been performed. However, while Crazy-Kurt was running on a demo-line at PDC, we collected a corpus of calls and transcribed 86 dialogues distributed over three of the system iterations. Some of the dialogues were performed by children. Although the system was simple, we learned a number of useful things to be exploited in a new version of the system. In particular we learned about user strategies when users talk to Kurt and what users perceive as entertaining.

Users often start conversation in what would seem to be a standard way of getting basic information on the interlocutor. This is probably triggered by the system's initial question but also by the fact that users know little or nothing about Kurt when they call him. The situation is much like a dinner party where you sit next to a person you never met before. The system starts by saying: "Hello it's me. What do you want to know about me?". In reply most users start by asking who the system is, cf. the dialogue in Table 4. There are also initial questions concerning e.g. what he knows about, his age, where he is, and how he is.

Many of the questions and comments provided by users during calls actually concern the system ("me"). Topics include – in addition to identity and age – where he works, where he lives, his childhood, wife, children, health, hair, eye colour, glasses, whether he smokes, is homosexual, or is a vegetarian. There are also a number of questions and comments including adjectives describing the system, e.g. stupid, clever, and handsome. Moreover, there are several questions/comments concerning his likes and dislikes regarding e.g. alcohol and food, football, music, work, and sex.

Kurt is very focused on himself and this is probably part of the reason why many questions/comments concern Kurt. However, it is also likely that the user is seeking for common ground, i.e. something to talk to Kurt about in more detail. This is often what one does when sitting next to a new person. There are only few examples in the corpus of the user offering information about himself, such as "I play Diablo" or "I live in Fredensborg".

It seems clear that Kurt should help the user more regarding what are reasonable conversation topics, e.g. by suggesting topics himself early on, such as "do you like vegetables/peas/aubergines/…" or "I have blue hair. What colour is yours?".

A large group of the input utterances relate to the dialogue and to what the system says. They may be viewed as a kind of meta-communication. They are often triggered by the too small grammar but are a major part of the entertainment. Many children were really laughing when they heard what the system recognised, no matter if correct or not, cf. Table 3. When misunderstood, users would sometimes try to correct the system. In other cases they provided long input just to see what the system recognised and get a good laugh.

The system's grammar definitely needs to be enlarged, what we also knew in advance. The small grammar means that the dialogue too often ends in system misunderstandings. Although these may be fun when the system tells what it understood, the user may also feel that there are too many of them and that there ought to be at least some progress in conversation. It is somewhat difficult to have a coherent dialogue with the system for more than a few turns.

Table 3. An excerpt of dialogue 2002-04-05-0-4-14, translated to the English.

No.	Utterance
U42.1	Yes you are a fool.
S43.1	Bah.
S43.2	Listen to yourself: "`You are a fool`".
S43.3	No.
S43.4	You are challenged regarding intelligence, poor you.
U43.1	Okay, twit. [laughter]

Clearly an enhancement of Kurt's background knowledge about himself is needed as well. His limited knowledge is also somewhat prohibitive for a longer coherent dialogue. Moreover, he should be able to tell in a comprehensible way what he can chat about – at least when asked and maybe also if there is a problem with the dialogue. He may e.g. be equipped with a few interests which he can discuss in some detail.

At a point we re-recorded the output phrases, now using a 14 years old teenage boy instead of the original 40 years old male. This triggered new kinds of input, like "Who are your parents?". This emphasizes that in creating a character, all system features affect user behaviour and should be designed to fit together.

In the input quotations the synthetic voice would say the exact words that were recognised, e.g. "you said '`you are stupid`'" which resulted in responses like "no, you are stupid" instead of the design expectation "yes (I said (you are stupid))".

Table 4. Dialogue 2002-04-08-0-0-0, translated to the English. Legend: S system. U user, CL Change level, P Preference, Sat Satisfaction, 0 neutral, -1 cross.

No.	Utterance	CL	P	Sat	Act type
S1	Hello, it's me. What do you want to know about me?			0	Introduction
U1	What is your name?	0	0		
S2	Eh, try again.			0	Prompt
U2	Who are you?				
S3.1	Well			0	Change marker (neutral)
S3.2	Do you really mean "`Who are you`"?			0	Explicit feedback
U3	Yes.				
S4.1	I thought so.			0	Change marker (neutral)
S4.2	I'm Kurt the Kurterman, the toughest fruit eater in this country.			0	Answer
S4.3	What else do you want to know about me?				Prompt
U4	You are stupid.	0	0		
S5	Did you say "`You are blue`"?			0	Explicit feedback
U5	No, you are stupid.	-1	0		
S6.1	Hn.			-1	Change marker (cross)
S6.3	You did say "`You are stupid`" didn't you?			-1	Explicit feedback
U6	Yes.				
S7.1	If only you'd find half a snail in your salad.			-1	Change marker (cross)
S7.2	No.			-1	Answer
S7.3	Holluja, what a fool you are.			-1	Prompt (passive)
U7	Bye.				
S8	What? Are you leaving? We just had great fun.			-1	Closing

4 Related work

Picard [10, p.2] notes that evidence is mounting that "emotional skills … are essential for natural communication with humans" and coins the area *affective computing*. However, there is only little work on spoken dialogue for entertainment and including emotion representation. The very thorough book by [2] notes on p. 246 that "very little is known about effective design techniques for personified interfaces", and itself only deals with "professional interfaces", and not at all with affection or emotion.

Affective computing is concerned both with recognising, representing and acting upon, and expressing emotion.

Batliner et al. [3] address *recognition* of user emotions and list several sources for this purpose including prosody, repetitions, reformulations, dialogue act sequences, facial expression, and words. They show that prosody alone is a fairly good indicator of emotion, but that adding repetition and reformulation cues may improve the error rate by one third. Contemporary commercial recognisers do not support prosodic features, but grammatical cues may be used. We have used this in [7], in order to apologise or provide more explicit help.

Representation of emotion is common in the agent literature, e.g. [1, 9]. Davis' [5] emotion engine that models each emotion by a cellular automaton, is a good example. Scheutz and Logan [11] describe three basic architectures for affective systems: The *reactive* architecture where input directly triggers actions; the *affective* architecture where explicitly represented emotions control the state; and the *deliberative* architecture where alternative actions are considered before any action is taken. They show by experiments that the affective architecture is most effective, unless an advanced deliberative architecture is used, and even then the affective approach is better in a number of situations. Deliberative architectures are difficult to construct so that they sufficiently take into account all situations. The architecture of the Crazy-Kurt system would be categorised as affective.

The output generation we use for *expressing* emotion is simple (cf. Section 2.6). Walker et al. [12] describe a more elaborate model based on speech acts as well as social variables and face. Also, Gricean maxims are commonly used as design principles for task-oriented dialogue [4]. In Gricean literature it is repeatedly stated that to *flout* maxims (i.e. to violate them on purpose) can be used to achieve effects like sarcasm and humour.

Techniques based on recognition confidence scores are often used [2, 6, 8]. However, the optimal use of score values is still only investigated in little detail.

Finally, there is a clear relation to Weizenbaum's Eliza [13] and to the modern web chat-robots based on written input. However, Kurt is significantly different in at least two ways: (i) Kurt performs spoken conversation, and (ii) Kurt has a personality which means that there is a development over time in a conversation with him.

5 Conclusion and next steps

We have described the present version of a non-task-oriented conversational spoken dialogue system whose affective character, Crazy-Kurt, is meant to entertain users who call him and chat with him about himself, primarily about his preferences of food. In particular we have discussed techniques used in the modelling of Kurt's personality and how his changing mood is reflected in his output.

Despite several shortcomings it is our impression from experiments that users – and not least children – generally find the system entertaining. In particular they seem to enjoy the system's repetition of what it understood and to provide some kind of insulting input to see how this affects Kurt and his output.

We are fully aware that the present version needs improvement in several ways. As a first step we want to extend the grammar coverage which presently is quite poor and results in too many misunderstandings. Kurt even breaks the design rule that a system should understand its own language.

But extending the grammar is not sufficient. We also want to extend Kurt's knowledge about himself since there are too many questions he cannot answer. An idea would be to give him quite detailed knowledge about one or more of his interests. As part of this we would like to investigate techniques which may help in achieving that users naturally stay within Kurt's domain of knowledge.

We consider enabling Kurt to ask questions about the user to obtain a more conversational interaction. Moreover, we would like to enable understanding of other kinds of user speech acts, such as swearing, to obtain a more varied dialogue and a more complete conversational character.

The modelling of personality is so far fairly simple. We would like to experiment with the addition of a number of new parameters, including patience, balance, self-esteem, and pessimism/optimism. Which weight to give to each of the personality parameters included will depend on the personality one wants to model. For example, patience may be modelled via the three input events too long, silence, and barge-in. An impatient person is quite likely to interrupt the interlocutor and take the initiative. A patient person, on the other hand, tends to listen until the interlocutor has finished his turn. Thus an impatient system may have a low limit for too long input. Also an impatient system may only accept very short silence from the user before it takes the initiative and perhaps directly asks why the user does not answer. The system's degree of patience may decrease in case of misunderstandings and if the user interrupts (via barge-in[1]) too often since it is annoying to be interrupted many times. If the system's patience falls below a certain threshold it may even disable barge-in and tell the user that now he has to listen to the system for a moment.

Other ideas we would like to consider include entirely new kinds of interaction patterns, such as letting Kurt hand over the phone to a colleague if he becomes too

[1] Though barge-in is supported by most speech platforms, the event is not explicitly accessible to the dialogue designer. In Kurt as well as in task-oriented dialogue it is important in meta-communication to know how much of the phrase and information was actually output. In HDDL it is possible to circumvent the lack of this accessibility using intricate programming.

bored or realises he is on the verge to be disclosed as incompetent. Or we may have sessions where the system detects that it was called before from the same number or by the same person. This is realisable via storing the telephone A-numbers and, on certain platforms, via speaker recognition techniques.

The next version of Kurt will include many of the above extensions. We will test how well they work in practice and experiment with the weighting between different personality parameters as well as with the entertaining features of the system.

In summary, today's speech technology opens up for new classes of applications that exploit affective and conversational dialogue, either in terms of new kinds of non-task-oriented applications in the "-tainment" area, i.e. in entertainment, edutainment and infotainment, or in terms of – in more subtle disguise – improvements regarding dialogue robustness in current task-oriented systems.

References

[1] *Proceedings AISB 2001: Agents and Cognition*, University of York, March 2001.

[2] Bruce Balentine and David P. Morgan: *How to Build a Speech Recognition Application: A Style Guide for Telephony Dialogues*. EIG press, 2. ed., 2001.

[3] Anton Batliner, Richard Huber, Heinrich Niemann, Elmar Nöth, Jörg Spilker and Kerstin Fischer: The Recognition of Emotion. In W. Wahlster (editor): *Verbmobil: Foundations of Speech-to-Speech Translations*, pages 122-130. Springer, New York, Berlin, 2000.

[4] Niels Ole Bernsen, Hans Dybkjær and Laila Dybkjær: *Designing Interactive Speech Systems. From First Ideas to User Testing.* Springer Verlag 1998.

[5] Darryl N. Davis: Multiple Level Representation of Emotion in Computational Agents. In [AISB 2001].

[6] Hans Dybkjær, Niels Ole Bernsen and Laila Dybkjær: Inferential Issues in a Spoken Language Dialogue System. In *FQAS'96*, Roskilde University, Denmark, 1996.

[7] Hans Dybkjær and Laila Dybkjær: Experiences from a Danish Spoken Dialogue System. *Second Danish HCI Research Symposium*, Copenhagen, Denmark, November 2002, 15-18.

[8] Andreas Kellner, Bernd Souvignier, Thomas Portele and Petra Philips: Dialog System. *US patent application 20020065651*, 2002-05-30.

[9] Ana Paiva (Ed.): *Affective Interactions. Towards a New Generation of Computer Interfaces. Springer Lecture notes in artificial intelligence*, vol. 1814, 2000. VIII + 235 pp. From workshop with same title in conjunction with 1999 I3 annual conference, Siena.

[10] Rosalind W. Picard: *Affective Computing.* MIT Press, 1997, first paperback 2000. XII + 292 pp.

[11] Matthias Scheutz and Brian Logan: Affective vs. Deliberative Agent Control. In [AISB 2001, 1-10].

[12] Marilyn A. Walker, Janet E. Cahn and Stephen J. Whittaker: Improvising Linguistic Style: Social and Affective Bases for Agent Personality. In *Proc. of the conf. on autonomous agents, Agents'97*, Marina del Rey, California, USA. 1997, 96-105.

[13] Joseph Weizenbaum: ELIZA-A Computer Program For the Study of Natural Language Communication between Man and Machine. *Communications of the ACM*, Volume 9, Number 1, January 1966, 36-35.