SEMAINE

THE SENSITIVE AGENT PROJECT

D4a
Updated demonstrator of the Dialogue Manager

Date: 22 December 2009

Dissemination level: Public

ICT project contract no.	211486
Project title	**SEMAINE** **Sustained Emotionally coloured Machine-human Interaction using Nonverbal Expression**
Contractual date of delivery	*31 December 2009*
Actual date of delivery	*22 December 2009*
Deliverable number	D4a
Deliverable title	Updated demonstrator of the Dialogue Manager
Type	Demonstrator
Number of pages	16
WP contributing to the deliverable	WP 4
Responsible for task	Dirk Heylen (d.k.j.Heylen@ewi.utwente.nl)
Author(s)	Elisabetta Bevacqua, Dirk Heylen, Mark ter Maat, Catherine Pelachaud, Etienne de Sevin
,EC Project Officer	Philippe Gelin

Table of Contents

1 Executive Summary

Sensitive Artificial Listeners (SAL) are virtual dialogue partners who, despite their very limited verbal understanding, intend to engage the user in a conversation by paying attention to the user's emotions and non-verbal expressions. The SAL characters have their own emotionally defined personality, and attempt to drag the user towards their dominant emotion, through a combination of verbal and non-verbal expression.

The SEMAINE system 2.0 is the first public demonstrator of the fully operational autonomous SAL system based on audiovisual analysis and synthesis. The present report is part of a group of reports describing various aspects of the SEMAINE system 2.0. The full list of reports is available from http://semaine.opendfki.de/wiki/SEMAINE-2.0.

This report describes the progress made on the Dialogue Management part of the system. The Dialogue Manager components are responsible for making sure that the conversation and interaction of the human with the virtual agent takes place. To do this, the dialogue manager needs to manage a number of things, such as the interpretations of the user behaviour, the turn taking behaviour, the backchanneling behaviour, and the utterance selection of the agent.

The current version of the Dialogue Manager, remains rather ad hoc, as it can rely only on a limited number of input features. As richer input becomes available and the generation modules are further optimized, the Dialogue Manager modules can be developed further to the next level, with full testing becoming an option as well.

2 System description

Dialogue management proceeds in several steps. First there are modules interpreting the input data that is received (modules in WP3) trying to find meaning in the user's behaviour. These interpretations are then used to find the right places to backchannel, and to find the best time to start speaking. When the decision is made to produce an utterance, an appropriate response has to be found that fits the current context. In the current version of the interpretation components remain basic as the different input modules have become available only in the latter part of the year and are still very much under development.

The Dialogue Manager components are responsible for making sure that the conversation and interaction of the human with the virtual agent takes place. To do this, the dialogue manager needs to manage a number of things, such as the interpretations of the user behaviour, the turn taking behaviour, the backchanneling behaviour, and the utterance selection of the agent. Besides the turn taking module, the utterance selection module and the backchannel modules (discussed in separate subsections below), the following modules are part of the Dialogue Manager.

- Agent mental state interpreter – Keeps track of the agent's mental state, 12 variables used for choosing the type of backchanneling behaviour to perform (e.g. agree/disagree, belief/disbelief). At this moment it uses a certain baseline for each character, and it uses simple rules to modify the values. Most of these rules reward when the user is more like the character. For example, a raised arousal value when speaking to Spike will result in a higher value for agreement and liking. However, these rules are hand crafted and need to be updated next year.
- Emotion interpreter – A simple module, puts detected emotions in the user state if they exceed a certain confidence threshold.
- Head movement interpreter – Tries to interpret head movements. At the moment, a nod is interpreted as an agreement, and a shake is interpreted as a disagreement. These are put in the user state.

We will describe the turn-taking, utterance selection and backchannel module in turn.

2.1 Turn Taking

When trying to make conversations with the SAL system as natural as possible an important element is the turn taking in that conversation. One would want this to go smoothly. This means that SAL should not wait too long after the user is finished, but neither should react too soon as this increases the chance that it misinterprets the turn-end and overlaps the user when it does not want to.

In the previous version of SAL the turn taking mechanism was implemented in a very simplistic manner. The Dialogue Manager (DM) would wait until it detected at least two seconds of silence of the user, after which it would just start speaking. This gap is too big, however. It was chosen so as to be sure that the agent's speech would not overlap with that of the user. For fluent conversations however this is not acceptable.

Identifying the end of the user turn as it happens is a non-trivial task. The problem becomes the more relevant as there is also a delay on the generation side of the system (the time it takes to create the animation and the speech). In order to respond quickly the system has to be able (ideally) to *predict* the end of the user turn. That is why we have worked on end-of-turn prediction in the context of WP3. We will continue to work on this in the next year as with the input modules that are becoming available it might be possible to make more accurate predictions having recognized relevant turn-yielding cues.

In the current SAL system the turn taking component a different approach is taken in which the system is not only reactive but also proactive. The turn-taking component does not simply wait for the user to finish the turn, but also keeps track of its own intention to speak. It constantly calculates the value of this intention; the eagerness of SAL to start speaking. The fact that the user finishes his or her turn of course is part of this value, but there are also other factors. The value is calculated currently as the sum of the following elements.

- User silence time – a value between 0 and 100 which increases over time when the user is silent
- Emotion – a value between 0 and 80 with 10 points for every detected emotional event (for example a peak in the arousal)
- User speaking time – a value between 0 and 30 that increases over time when the user is speaking (reaches its max after 30 seconds). This is to stimulate the agent to take the turn when the user is speaking for a longer period.
- Agent end wait time – a value between -100 and 0 that starts at -100 after the agent finishes its turn, and for the next two seconds rises to 0. This makes sure that the agent does not start too soon after its own utterance.
- user not responding – a value between 0 and 100 that starts rising when the user does not start talking after the agent finished its turn. It starts after 2 seconds and rises to 100 in 4 seconds unless the user starts speaking.

SAL currently takes the turn when the intention-value exceeds a certain threshold. These thresholds are made different for each of the different characters so as to define typical traits of the characters. For instance, an aggressive Spike can be assumed to want to speak faster and interrupt the user, whereas the more depressive Obadiah character is much more passive. For a study conducted on turn-taking and personality see Deliverable D0b and Ter Maat and Heylen (2009).

The turn-taking principles remain ad hoc and have not been fully tested. With a more stable system, more input variables[1], a better recognition performance and faster output generation, it will be possible to test the rules and procedures in real conversations and update the implementation accordingly.

2.2 Utterance Selection

In order for a conversation to be coherent, the utterances of the agent need be carefully selected. To select a right response the agent can take into account several variables in the current system, such as keywords detected or the arousal state of the user. To make this process transparent the utterance selection consists of a number of simple models which each look at one aspect of the utterance selection.

Currently, the following modules are implemented.

[1] For example, when the user stops speaking and also turns his or her head to look at the agent this is a very strong cue that he or she wants the agent to start speaking. Also, when the user looks up with a raised eyebrow this is a good indication that he or she is trying to remember something and is not yet finished with his or her turn. We are currently looking for such cues in the data and putting down requests for the input module developers to produce recognizers for these cues (see also Deliverable D0b).

- The After Silence module suggests responses to occur after a long period of user silence. It includes responses such as 'are you still there?'. These sentences are specified in the utterance config-file.
- The Linking Sentence module suggests responses based on specified linking sentences. These are sentence pairs which can be linked by a typical user response. For example, when the agent asks 'Have you done anything interesting lately?', and the user responds with a short answer with an agreement in it, a linking sentence could be 'You did? Great! Please tell me about them.'. The linking sentences (and the requirements of the user's response) are specified in the utterance config-file.
- The Content Module suggests responses based on the detected keywords. This module is based on annotated transcriptions of the WOz recordings made in the Humaine project (Cowie et al. 2008). These transcriptions were annotated on certain high-level features such as 'talking about past', 'talk about own feelings', 'agree/disagree', etc. Using this data an interpreter was written which uses detected keywords to find those features, and in the utterance selection model these features are then used to find a good response.
- The Arousal module suggests responses based on a very high or a very low arousal of the user (for example, Prudence might say 'Don't get too excited' after detecting high arousal). These sentences are specified in the utterance config-file.
- The Backup Responses module suggest some generic responses that fit in most of the cases. These sentences are also specified in the utterance config-file.

All modules return a list of suggestions plus a quality value for every suggestion. When all modules have made suggestions, the quality values are updated based on their recent occurrence. Responses that have only recently been used by the agent will get a lower quality. After this step the response with the highest quality is chosen.

Most of the models (all except the content-model) are still hand crafted, based on intuition of what a good response is, and on the basis of the input modules that are currently available. They should be seen as slot-fillers rather than as definitive modules. Several steps are underway to improve the system in the next year. First of all, the existing models will be fully tested with users. This will be done by selecting conversation fragments with a response selected by a model, and letting these responses be rated by human. Given that the input the dialogue manager has available, is very limited, a comparison could be made with a W0z system, where the Wizard needs to select utterances based on the same minimal input. But besides this the system also needs a model that is based on real data. For this, also other WOz data will be gathered.

The user turn is analysed for specific content-features in UtteranceInterpreter. This module receives its input from:
•*semaine.data.state.user.emma* - for the detected keywords
•*semaine.data.state.user.behaviour* - for detected head nods and shakes
Every time a new input is received it is analysed and send to *semaine.data.state.user.behaviour*.

The selection of the agent utterance is done in UtteranceActionProposer. This module gets its input from:
•*semaine.data.state.agent* - for the turn taking intention of the agent, determined in the TurnTakingInterpreter
•*semaine.data.state.user.behaviour* - for the user speaking state, detected emotions, and the analysed keywords
•*semaine.data.state.context* - for character switches and user (dis)appearances

•*semaine.callback.output.audio* - to get start and stop messages of executed agent-utterances
If the agent has the intention to take the turn, it will go through all models, select an utterance and send this to *semaine.data.action.candidate.function*. Also the agent turn-state is changed, and this is send to *semaine.data.state.dialog*. If the character will be changed, then this change will be send to *semaine.data.state.context*.

2.3 Listener intent planner and listener action selection

2.3.1 Listener intent planner

The Listener Intent Planner computes the agent's behaviour while being a listener conversing with a user. To display believable listener behaviour, the system must be able to: decide *when* a backchannel signal should be emitted and select *which* communicative intentions the agent should transmit through the signal. The algorithm steps that perform these tasks are shown in Figure 1. The system receives in input information about the user's behaviour (STEP 0); then it triggers a backchannel (STEP 1); the backchannel is actually emitted if certain conditions are satisfied (STEP 2). Afterwards all the possible types of backchannels are generated (STEP 3). Three types of backchannel behaviours can be provided: reactive backchannel, response backchannel and mimicry. STEP 1 and STEP 2 compute *when* a backchannel signal should be emitted, while STEP 3 calculates *which* type of backchannel signals the agent should perform.

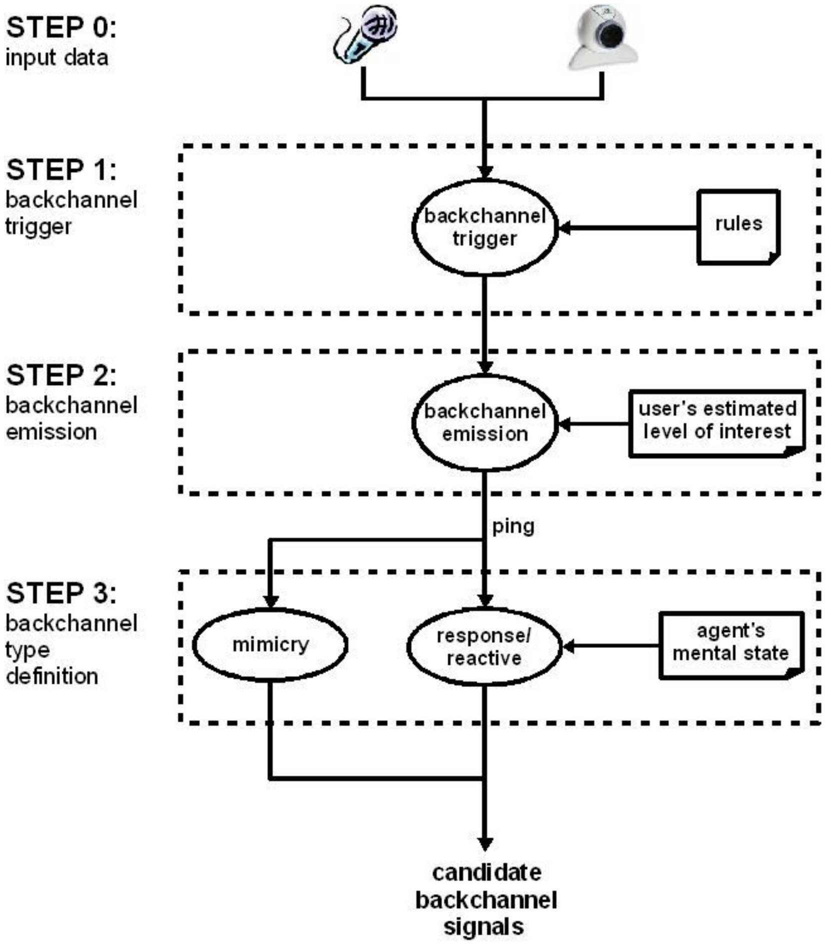

Figure 1: The four steps of the Listener Intent Planner algorithm.

STEP 0: input data
To identify those behaviours of the user that could elicit a backchannel from the agent, user's verbal and non-verbal behaviours are continuously tracked through a video camera and a microphone. Audio and visual applications can be connected to our system to provide information about head movement, facial actions, acoustic cues like pauses and pitch variation in the user's voice. All information is provided at a certain level of interpretation and described in terms of behavioural signals; for example the application that analyzes facial actions is able to interpret lip movement to recognize a smile. In the SEMAINE project, the Listener Intent Planner has been connected with the video analysis applications developed in the Imperial College and with the audio analysis applications developed in TUM.

STEP 1: backchannel trigger
Research has shown that there is a strong correlation between the triggering of a backchannel signal and the verbal and non verbal behaviours performed by the speaker (Maatman et al., 2005, Ward and Tsukahara, 2000). Models have been elaborated that predict when a backchannel signal can be triggered based on a statistical analysis of the speaker's behaviours (Maatman et al., 2005, Morency et al., 2008, Ward and Tsukahara, 2000). We use a similar approach and we have fixed some probabilistic rules to prompt a backchannel signal when our system recognizes certain speaker's behaviours; for example, a head nod or a variation in the pitch of the user's voice will trigger a backchannel with a certain probability. The probabilities of triggering a backchannel signal are set based on studies from the literature. For example, Ward and Tsukahara (Ward and Tsukahara, 2000) looked at the low pitch and when it lasts 110 ms in the speaker's speech, it can be a good predictor for providing a backchannel signal. From their finding we associated a high probability (0.95) to trigger a backchannel to the user's event “silence”, that is when the acoustic analyzer finds a region of low pitch in the user's speech.
The rule component contains the probability to generate a backchannel and a priority value that helps the Action Selection module to determine which backchannel will be actually triggered when several user's behaviours satisfy more than one rule at the same moment. At present priorities are set based on literature and on observation studies. The rules are defined through an XML-based language and are written in an external file uploaded at the beginning of the interaction. By using such a type of language, the set of rules can be easily modified or extended. To take into account user's signals analyzed by new applications, we can add new rules in the external file without modifying the source code. Moreover, we can easily modify the probability associated to those user's behaviours that can trigger a backchannel signal.

STEP 2: backchannel emission
The *Backchannel Emission* module, decides whether a backchannel request is actually generated. We implemented such a step to add some variability to the backchannel emission frequency. As a first approach, we decide to base the computation of the backchannel emission frequency only on the *user's estimated interest level* towards the interaction. We define interest as an emotional state linked to the user’s goal of receiving and elaborating new and potentially useful knowledge (Peters et al., 2005). The user's estimated interest level is a value between 0 (minimum interest) and 1 (maximum interest). We consider the user's estimated interest level as an indicator of the successfulness of the interaction: when the interest level decreases it may be a sign that the user

might want to stop the conversation (Schegloff and Sacks, 1973), consequently the probability that the agent provides a backchannel decreases.
When a user's behaviour satisfies one of the rules, a backchannel is triggered. The Backchannel Emission module sends a *ping* to “wake up” the modules charged with the generation of the backchannel signal, with a probability computed from the user's estimated interest level.

STEP 3: backchannel type definition
The *Response/Reactive* and the *Mimicry* modules are charged with the definition of the type of the backchannel to emit. We take into account three types of backchannel signals: reactive, response and mimicry. Our agent can emit reactive backchannels that are signals derived from perception processing: the agent reacts to the speaker's behaviour or speech, generating automatic behaviour. Moreover, our agent can provide response backchannels that are signals generated by cognitive processing: the agent responds to the speaker's behaviour or speech performing a more aware behaviour. The Response/Reactive module generates both types of backchannel signals. The Mimicry module is in charge of the generation of a particular type of backchannel signals: signals of mimicry, that is the copy of certain user's behaviours. We are interested in this type of backchannels since researches have shown that mimicry helps to make the interaction an easier and more pleasant experience, improving the feeling of engagement (Chartrand and Bargh, 1999, Cassell et al., 2001, Warner et al., 1987, Chartrand et al., 2005). In order to compute the backchannel to display, information about what the agent “thinks” of the speaker's speech is needed. This information is provided in the agent's *mental state* that describes whether the agent agrees or not, believes or not and so on. We define the mental state as a set of communicative functions the agent wishes to transmit during an interaction. For each communicative function the value of the importance the agent attributes to it is defined. Such a value is a number between 0 and 1, where 0 represents the minimum importance whereas 1 indicates that the agent gives to the corresponding communicative function the maximum importance. We consider twelve communicative functions, a subset chosen from the taxonomies proposed by Allwood et al. (Allwood et al., 1993) and by Poggi (Poggi, 2007): agree, disagree, accept, refuse, believe, disbelieve, interest, not interest, like, dislike, understand and not understand. To each communicative function is associated a set of behavioural signals to convey the given function.

Response/Reactive module
The Response/Reactive module uses the information in the agent’s mental state to compute the appropriate backchannel. This module must generate a backchannel when a “ping” from the Backchannel Emission module is received. Firstly, the Response/Reactive module looks for all the communicative functions in the agent's mental state that have a value of importance higher than zero. Then, if at least one communicative function responds to these criteria, this module generates a response backchannel. The backchannel is written in a message in FML-APML format and it contains all the communicative functions that have a value of importance higher than zero. It will be up to the Behaviour Planner to select the adequate behaviours to display for each communicative function according to their importance. If no information is given in the agent's mental state, that is if any communicative function has a value higher than zero, this module generates a reactive backchannel: an automatic reaction to the user's behaviour that shows simply contact and perception. This type of backchannel is translated in those typical signals, like head nods and raise eyebrows, that have been studied in the literature (Allwood and Cerrato, 2003, Cerrato and Skhiri, 2003, Cerrato, 2002, Poggi, 2003).

Mimicry module

When fully engaged in an interaction, mimicry of behaviours between interactants may occur [Lakin et al., 2003]. It has been shown that mimicry, when not exaggerated to the point of mocking, has several positive influences on the interaction (Chartrand and Bargh, 1999, Cassell et al., 2001; Warner et al., 1987; Chartrand et al., 2005). This type of backchannel signals is generated by the Mimicry module. When the Backchannel Emission module sends a *ping*, the Mimicry module checks if it can generate a mimicry of the speaker's behaviour that has triggered the backchannel. This type of backchannel is written in BML language that allows the system to specify the signals to perform.

The Listener Intent Planner is implemented in the ListenerIntentPlanner component in the SEMAINE framework. It receives information from the Topics semaine.data.state.agent, semaine.data.state.user.behaviour, semaine.data.state.dialog, semaine.data.state.context. Reactive/response backchannels are sent as FML file to the Topic semaine.data.action.candidate.function and mimicry as BML to the Topic semaine.data.action.candidate.behaviour.

2.3.2 Listener Action selection

The Reactive/Response Backchannel module and the Mimicry module can generate backchannels that are potentially conflicting at the behaviour level. For example, the Mimicry module could generate a head nod to mimic the user's head movement, whereas the response backchannel module could generate a head shake determined by the communicative function “disagree”. There is a conflict between both head signals and as just one signal can be actually displayed, a choice has to be done (de Sevin, 2009).

The user's estimated interest level, estimated by user’s head direction (computed within WP3) is modelled by a value between 0 (minimum interest) and 1 (maximum interest). It can be an indicator of the successfulness of the interaction (Peters, 2005). We use this variable to vary the number of backchannels that are emitted. For example, when the interest level decreases, it can be interpreted as a sign that the user might want to stop the conversation (Schegloff and Sacks, 1973), consequently the probability that the agent provides a backchannel decreases.

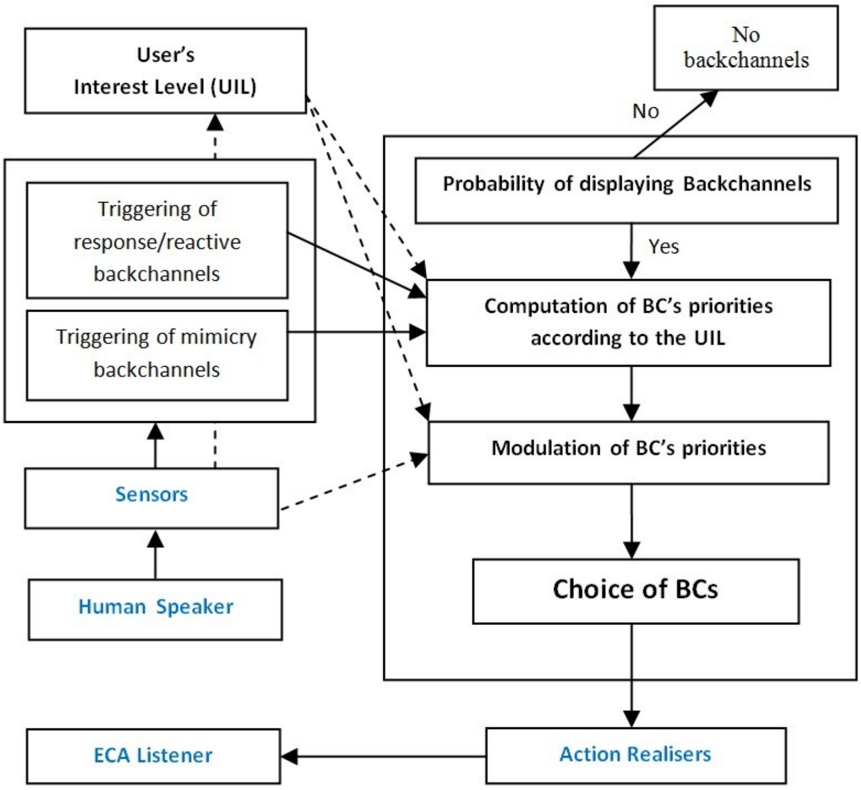
User’s Interest Level (UIL)
No backchannels
No
Probability of displaying Backchannels
Yes
Triggering of response/reactive backchannels
Triggering of mimicry backchannels
Computation of BC’s priorities according to the UIL
Modulation of BC’s priorities
Sensors
Human Speaker
Choice of BCs
Action Realisers
ECA Listener

Figure 2. Schematic view of the Backchannel architecture including a backchannel (BC) selection module

As the backchannel selection algorithm receives all the potential backchannels with their priorities from the action proposers without any choices in the action proposers (de Sevin, 2006), the algorithm proceeds as follows (see Figure 2):

- Computation of the probability of displaying backchannels for all received ones.
- Calculation of the priorities of BCs according to the user's level of interest (estimated by the agent)
- Modulation of the backchannel's priorities according to the estimated gaze of the user, the user's level of the disinterest or the phase of the interaction (begin, maintain or end).
- Selection of the most appropriate backchannels among possible conflicting ones to be sent to the Action Realisers.
- Wait until the chosen backchannel is finished to be displayed by the player before choosing another one. Backchannels and utterances which are received by the selection algorithm during this time are queued with their priorities.

Computation of backchannel's priorities according to the user interest level

The selection algorithm has to choose between two types of backchannels: mimicry and response(/reactive) backchannels (see Figure 1). Mimicry is chosen preferentially when the agent perceives that the user is very interested in the interaction so that the agent can show its high engagement in the interaction (Gratch et al., 2007; Thiebaux et al., 2008). The response backchannels, showing the communication intentions of the agent, are used when the ECA detects that the user looses interest in the interaction. It is also used to encourage the user to be interested in the interaction (Goodwin, 1981).

We have designed 2 polynomial functions, one for each type of backchannels (see Figure 3), for calculating their priorities according to user's interest level and the rule priorities of the backchannels coming from the triggering modules.

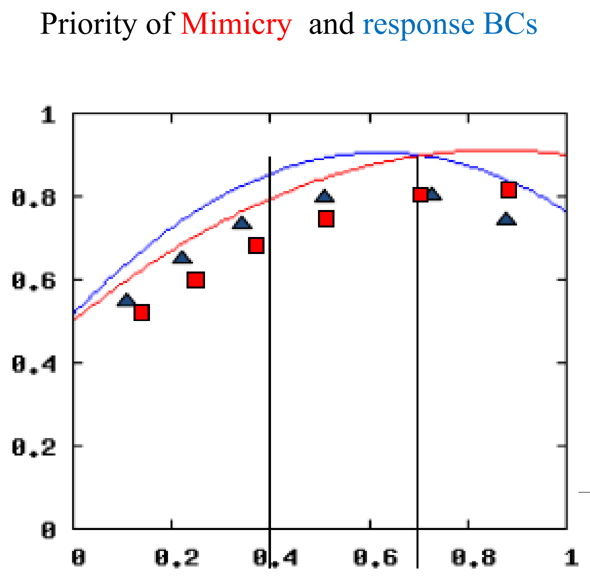

Figure 3. Polynomial functions to calculate the priority of mimicry and response backchannels according to the perceived interest level of the user. PLUI: Perceived Level of User's Interest

We define the relation between backchannel priorities and the user's interest level as follow. When the ECA estimates that the interest level of the user is near to the maximum (PLUI > 0.75), mimicry is chosen preferentially. This is somehow linked to the notion of engagement (Sidner et al., 2004). If the agent detects that the user begins to be less interested (PLUI < 0.75), response backchannels are chosen preferentially in order to keep the user interested or to increase the interest of the user if it is the beginning of the interaction. After a while when the agent detects that the user begins to be disinterested (PLUI < 0.4), the agent considers that the interaction is ending and stops progressively doing backchannels.

These polynomial functions are used to normalize all backchannel priorities according to the interest level of the user. This normalisation process allows us to compare BC priorities and to select one of them. The value of these priorities can vary afterwards according to the context of the interaction such as if the user is looking at the ECA or if he begins to be disinterested.

The selection is event-based and is done in real-time. If backchannels are triggered, then a choice is made. However when the ECA is already displaying a backchannel, no choices are made. The algorithm waits until the agent finishes displaying a BC before selecting another one to be displayed. But if the selection algorithm receives some requests to select other backchannels to be displayed while the agent is displaying a BC, these requests are queued and used during the next selection pass. Finally, the selection algorithm chooses the most appropriate backchannels based on the priority values according to the user's interest level and the context of the interaction.

The Listener Action Selection is implemented in the Action Selection component in the SEMAINE framework. It receives candidate FMLs from the Topic semaine.data.action.candidate.function and BMLs from semaine.data.action.candidate.behaviour coming from Action Proposers. It also uses information from the Topics semaine.data.state.agent, semaine.data.state.user.behaviour, semaine.data.state.dialog, semaine.data.state.context and semaine.callback.output. Selected FMLs are sent to the Topic semaine.data.action.selected.function and selected BMLs to the Topic semaine.data.action.selected.behaviour.

3 License and availability

The dialogue components are available as part of the SEMAINE 2.0 system, under the LGPL license.

Greta is available from http://www.tsi.enst.fr/~pelachau/Greta/. It is licensed under GPL licence. Greta and the four facial models are part of the SEMAINE 2.0 system release.

References

(Allwood et al. 1993) Allwood, J, Nivre, J, and E., A. On the semantics and pragmatics of linguistic feedback. Semantics, 9(1), 1993.

(Allwood and Cerrato 2003) Allwood, J. and Cerrato, L. A study of gestural feedback expressions. In Paggio, P., Jokinen, K., and Jónsson, A., editors, First Nordic Symposium on Multimodal Communication, pages 7-22, Copenhagen, 2003.

(Cassell et al, 2002) Cassell, J., Nakano, Y., Bickmore, T., Sidner, C., and Rich, C. Non-verbal cues for discourse structure. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pages 114-123. Association for Computational Linguistics Morristown, NJ, USA, 2001.

(Cerrato, 2002) Cerrato, L. A study of verbal feedback in italian. In NORDTALK Symposium on Relations between Utterances, pages 80-97, Copenaghen, 2002.

(Cerrato and Skhiri, 2003) Cerrato, L. and Skhiri, M. Analysis and measurement of head movements signalling feedback in face-to-face human dialogues. In Paggio, P., Jokinen, K., and Jónsson, A., editors, First Nordic Symposium on Multimodal Communication, pages 43-52, Copenaghen, 2003.

(Chartrand and Bargh, 1999) Chartrand, T. and Bargh, J. The Chameleon Effect: The Perception-Behavior Link and Social Interaction. Personality and Social Psychology, 76:893-910. 1999.

(Chartrand et al. 2005) Chartrand, T., Maddux, W., and Lakin, J. Beyond the perception-behavior link: The ubiquitous utility and motivational moderators of non-conscious mimicry. The new unconscious, pages 334-361. 2005.

(Douglas-Cowie et al. 2008) E. Douglas-Cowie, R. Cowie, C. Cox, N. Amir, and D. Heylen. The sensitive artifcial listener: an induction technique for generating emotionally coloured conversation. In LREC2008, Marocco, May 2008.

de Sevin E. An Action Selection Architecture for Autonomous Virtual Humans in Persistent Worlds. PhD. Thesis. VRLab EPFL. 2006.

(de Sevin and Pelachaud, 2009) de Sevin, E. and Pelachaud, C.: "Real-time Backchannel Selection for ECAs according to User's Level of Interest". In Proceedings of Intelligent Virtual Agents 2009, IVA'09, Amsterdam, Holland. 2009.

(Gratch et al. 2007) Gratch, J., Wang, N., Gerten, J., Fast, E., and Duffy, R. Creating rapport with virtual agents. In et al., C. P., editor, 7th International Conference on Intelligent Virtual Agents, Paris, France, 2007.

(Goodwin, 1981) Goodwin, C.: Conversational Organization: Interaction between Speakers and Hearers. Academic Press. 1981.

(ter Maat and Heylen, 2009) ter Maat, M. and Heylen, D.K.J. Turn Management or Impression managementIn: Intelligent Virtual Agents, 9th International Conference, IVA 2009, 14-16 Sep 2009, Amsterdam, Netherlands. pp. 467-473. Lecture Notes in Computer Science 5773. Springer Verlag, 2009.

(Maatman et al. 2005) Maatman, R, Gratch, J, and Marsella, S. Natural behavior of a listening agent. In 5th International Conference on Interactive Virtual Agents. Kos, Greece, 2005.

(Morency et al. 2008) Morency, L.-P., de Kok, I., and Gratch, J. Predicting listener backchannels: A probabilistic multimodal approach. In Prendinger, H., Lester, J. C., and Ishizuka, M., editors, Proceedings of 8th International Conference on Intelligent Virtual Agents, volume 5208 of Lecture Notes in Computer Science, Tokyo, Japan. Springer, 2008.

(Peters et al. 2005) Peters, C., Pelachaud, C., Bevacqua, E., Poggi, I., Mancini, M., and Chafai, N. E. A model of attention and interest using gaze behavior. In Proceeding of IVA'05: Intelligent Virtual Agents, Kos, Greece, 2005.

(Peters, 2005) Peters, C.: Direction of attention perception for conversation initiation in virtual environments. In : International Working Conference on Intelligent Virtual Agents, Kos, Greece, pp.215-228. 2005.

(Poggi, 2003) Poggi, I. Mind markers. In Trigo, N., Rector, M., and Poggi, I., editors, Gestures. Meaning and use. University Fernando Pessoa Press, Oporto, Portugal, 2003.

(Poggi, 2007) Poggi, I. Mind, hands, face and body. A goal and belief view of multimodal communication. Berlin: J. Weidler, 2007.

(Schegloff and Sacks, 1973) Schegloff, E. and Sacks, H. Opening up closings. Semiotica, 8(4):289-327. 1973.

(Sidner et al. 2004) Sidner, C., Kidd, C., Lee, C., Lesh, N.: Where to Look: A Study of Human-Robot Interaction. In Press, ACM, ed. : Intelligent User Interfaces Conference, pp.78-84. 2004.

(Thiebaux et al. 2008) Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.: SmartBody: behavior realization for embodied conversational agents. In : Proceedings of 7th Conference on Autonomous Agents and Multi-Agent Systems, pp.151-158. 2008.

(Ward and Tsukahara, 2000) Ward, N and Tsukahara, W. Prosodic features which cue back-channel responses in English and japanese. Journal of Pragmatics, 23:1177–1207, 2000.

(Warner et al. 1987) Warner, R. M., Malloy, D., Schneider, K., Knoth, R., and Wilder, B. Rhythmic organization of social interaction and observer ratings of positive affect and involvement. Journal of Nonverbal Behavior, 11(2):57-74. 1987.