SEMAINE

THE SENSITIVE AGENT PROJECT

D4b

Final demonstrator of the Dialogue Manager

Date: 24 September 2010

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 August 2010*
Actual date of delivery	*24 September 2010*
Deliverable number	D4b
Deliverable title	Final demonstrator of the Dialogue Manager
Type	Demonstrator
Number of pages	13
WP contributing to the deliverable	WP 4
Responsible for task	Dirk Heylen (d.k.j.heylen@ewi.utwente.nl)
Author(s)	Mark ter Maat, Dirk Heylen
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.

This report is part of the series of reports describing the implementation of SAL in system SEMAINE-3.0. The software described, and the full set of reports, can be downloaded from http://semaine.opendfki.de/wiki/SEMAINE-3.0/.

This report describes the new approach to specifying the selection of utterances in SEMAINE system, version 3.0 and sketches the types of selection models that have been implemented. A first evaluation of parts of the new approach to dialogue management is also provided.

2 Functionality of the components

This section describes the functionality of the components in the SAL system. The possibilities to configure and reuse the components as parts of a research toolbox will be published as deliverable D7e in December 2010.

In Semaine 3.0, the way of selecting agent utterances (what to say when) has been changed radically in several ways as compared to the previous version. First, the way of specifying and coding the selection rules has changed. Instead of specifying the different selection models in the JAVA code, the selection models are now defined as XML templates, defining the preconditions and the effects of each rule. The new version provides an interpreter for these rules. Besides the change in the way the rules are coded, also the selection models themselves have changed. In the next sections we will first explain how the new template system works (section 2.1), and after that we will describe several selection models (section 2.2).

2.1 Template System

The rationale behind the template system is to have a flexible set of easily definable templates which specify exactly what kind of behaviour to perform under which circumstances. A template defines a behaviour rule: it describes the behaviour to perform when the rule is fired, the effects it has on the state of the conversation, and the current conversation state that is required to fire the rule. The dialogue system contains a list of these rules, and based on the current conversation state the `best' rule is selected and its behaviour is performed.

The current state of the conversation is stored in an Information State (IS). This object contains all necessary information about the conversation. The IS contains, amongst others, information about the user, for example behaviours detected by the input modules (head movements, speech, facial expressions, etc.) and emotions that were detected. It also contains information about the agent: the current character that is speaking, the speaking-state of the agent, and the history of spoken utterances.

Each of the templates has a set of preconditions that all need to be true to fire that template. These preconditions check certain conditions in the Information State. For example, a template could check whether the speaking-state of the user is 'listening', and the average F0-value of the last user-turn was greater than 400, and whether the list with detected content-features contains the feature 'agreement'. The templates are *checked* against the current state of the conversation every 50 ms. When checking the templates, the preconditions of all the templates are verified against the current IS. The second part of a template is the effects it has when it is executed. These include changes it will make to the Information State and an utterance to speak. The checking procedure only checks which of the templates preconditions are met, it does not execute the changes.

The result of the checking procedure is a list of templates that have all their preconditions met. In general, the templates can be divided in two categories. The first involves templates that do not specify actual behaviours of the agent (verbal or nonverbal). The second involves templates that do so. In further processing, the first step is to fire the templates that do not specify actual behaviour of the agent but that only make changes to the Information State. After that, a *selection algorithm* determines which of the templates that will start a certain agent behaviour will be executed. To improve this selection, every template defines the quality of its resulting behaviour with a number

between 0 and 1. A higher number means the resulting behaviour is of better quality, which means it will be chosen first before templates with a lower quality value. Also, utterances that were recently spoken by the agent will have their quality value temporarily lowered to decrease repetition. When a selection is made, the behaviour that is specified in the selected template (only one template that results in actual behaviour is chosen) is executed.

Here is an example template:

```
<template id="Poppy_RespondToSmile1" name="Poppy Respond to Smile">
   <preconditions>
      <compare value1="$Agent.character" value2="Poppy" />
      <compare value1="$UserState.AU.nrOfAU12" comparator="greater_than" value2="1" />
      <trigger value1="$Agent.speakingIntention" value2="want_turn" />
   </preconditions>
   <behaviour class="ResponseActionProposer" quality="0.5">
      <argument name="response" value="#Poppy126" />
   </behaviour>
</template>
```

This template requires the current character to be Poppy, the number of times that AU12 (a smile) is detected in the last user-turn to be greater than 1, and the agent should have the intention to take the turn. When these preconditions are met, and when this template is selected, it will start a new agent behaviour with response #Poppy126 which is the utterance "*Everything seems better when you can smile.*".

The use of templates was inspired by their use in Dipper (Bos et al. 2003).

2.2 Behaviour rules

In this section we will describe the different types of behaviour rules that are used in the Dialogue system. First, we will describe linking sentences, which use simple features from the user's speech to link a SAL sentences to the previous one. Next, rules that are based on audio features are described, and finally rules that respond to detected non-verbal behaviour are explained.

2.2.1 Linking sentences

In order to get more structured conversations that take some more context into account than the previous user utterance, the systems uses templates to link a response of an agent to a previous response. Manyof such linking responses were added to the system. For example, when Spike says 'Don't you ever get angry at the world?', and the user responds with a short, disagreeing ('no') answer, a good next response could be 'You've got to show people who's boss!'.

This kind of linking is an easy way to create a more structured conversation, without specifying much about the content. Several features of the user's turn are used to decide on the response. Currently, the features that can be detected are the length of the turn (short, normal or long) and agree/disagree (basically, if the user says yes or no). With these features, all agent utterances that are questions and that could be answered with yes or no have been extended with proper responses.

A list of example linking sentences can be found in Appendix A1.

2.2.2 Audio Based

Another type of templates use the features provided by OpenSMILE. Using machine learning techniques, utterance selection rules where extracted from the Solid SAL data and implemented in the templates. Details about this process can be found in section 2.3.

These rules are triggered by certain features in the audio. These can be raw audio features, such as the energy and the F0 frequency, but also higher level features such as valence, arousal, and interest.

For example, one rule for Poppy specifies that when at least 2 of the following conditions hold:
–Valence.nrOfHighEvents >= 2
–Arousal.nrOfHighEvents >= 2
–Valence.nrOfLowEvents <= 0
–Arousal.nrOfLowEvents <= 0
then a possible response would be “I'm so pleased for you.”.

2.2.3 Non Verbal Responses

A third type of template looks at the non-verbal behaviour of the user, and provides a respond to this. Because the non-verbal behaviour became available only recently, only a few of these templates are defined, but more could easily be added. These templates react to behaviour such as head movements, pitch contours, and facial expressions. An example is be a rule of Spike to say “Stop laughing!” whenever a laugh was detected.

2.3 Creating the AudioBased rules

As mentioned in section 2.2.2, we used machine learning techniques to create rules that base their response on the detected audio features. For this we used parts (recordings 4, 6, 9, 10, 11, 13, 15, and 16) of the Semaine Solid Sal data (Valstar et al. 2010). From this data, all the user turns (893) were extracted and presented to three annotators. The annotators were asked to choose three responses from the list of SAL utterances which they thought would fit best in response to the user turn. This resulted in a set of user turns and a list of possible responses that SAL could say.

Next, OpenSMILE was used to extract audio features from the user turns. This included low-level features such as the energy and F0, and emotional features such as valence and arousal. Using all this data, first the responses selected by the annotators were clustered into groups containing utterances with a similar meaning to decrease the number of possible responses classes for further processing. To do this, a Principal Component analysis was conducted on the audio feature data, to group the user-turns that are very similar. After this, the responses that belong to these clusters were assumed to convey the same meaning, and they were grouped.

At this stage, the data contained for each user-turn a set of audio features, and a list of response-groups that are possible SAL responses after this user-turn. By using several machine learning techniques, we wanted to get models that would learn from this data when to give what response. Optimally, it should be possible to describe these models with the templates that were described earlier. For this reason, we are using machine learning techniques that produce rules that can be converted to templates. Therefore, we used the Ripper technique, which produces rules, and the J48 technique, which produces a decision tree. To verify if a black-box technique (where the model is hidden from the user) would perform better, we also used a SVM. More information about this study can be found in (ter Maat & Heylen 2010).

3 Quality assessment

This section describes an assessment of the quality on the technical and component level. An assessment of the psychological quality of interactions with the overall system will be published separately, as deliverable report D6d, in December 2010.

3.1 Dialogue system evaluation

When evaluating the Dialogue management, there are two aspects that need to be evaluated.: The utterances it produces, and the timing of these utterances, that is, the turn taking. In this section, we will describe how to evaluate both aspects, first the turn taking, and finally the utterance selection.

3.1.1 Turn Taking

The dialogue system has to decide when to start speaking. It should not wait too long, but neither should it wait too briefly. To objectively evaluate the turn taking of the system, the time between the end of the user's turn and the start of the agent's turn has to be measured. The average duration of this pause should neither be too short nor too long. What durations are too short or too long should be defined beforehand. This could depend on the character. Also, the number of overlaps, where the agent starts talking while the user is still talking, should be counted. For some characters, for example Prudence, this number has to be as low as possible, since interrupting the user would be considered rude, which does not fit with Prudence. For other characters, for example Spike, it is fine if it interrupts the user sometimes, since this matches the aggressive attitude of Spike.

In order to evaluate this, the full system should be finished and conversations have to be held. The timings and overlaps can be calculated from the generated log-files. However, this can not be done at this moment in time, since the full system was just finished, and the experiments needed to gather this data will be gathered in the next few months. Relating turn-taking strategies (i.e timing of responses) and a character's personality was tested in an off-line, real-time study. See (ter Maat et al. 2010) for further details.

3.1.2 Utterance selection

Evaluating the utterance selection on a technical level is not a trivial task. An input component, for example the speech recognizer, has to recognize behaviour, and their technical performance is compared with a baseline, that is, the actual behaviour (in this case the words) that has to be recognized. However, the problem with the Dialogue Manager is that there is no baseline; it is very hard to objectively evaluate its performance since there is no objective data which contains user-turns and a list of all possible responses after this turn. Most of the evaluation will be based on the psychological evaluation where people subjectively rate the system. This evaluation will be done in a later phase of the project.

However, we would like to have an objective evaluation of the utterance selection as well. In order to compensate for this lack of a baseline, annotators used the Semaine data to suggest possible SAL responses for a set of user turns (see Section 2.3). This data was used to train classifiers, which were subsequently used to extract response selection rules.

There are some issues that need to be considered when using this data as a baseline for evaluation. First of all, the annotated responses are just three suggestions for each annotator, not a full list of

possible responses. So, a SAL response that was not suggested by the annotators is not necessarily a bad one . Second, because the annotators gave three suggestions for each user turn, each datapoint (user turn) has three (perhaps different) labels. This was partly countered by clustering the responses into groups (about four to six per character) and using the clusters as labels. This resulted in many data points now having the same labels, even though a lot of data points remained with different labels. This causes problems with the evaluation, because for example, if a data point has two labels, a suggested (by SAL) response will always have one matching label and one non-matching label. So, the more data points with multiple labels there are, the worse the evaluation will score based on simple matching. At this moment, in the annotation data the user turns have an average of 1.9 different labels. Of course it is possible to take these complications into account and to define more accurate evaluation metrics, but in the current, preliminary evaluations we did not do so.

The trained classifiers were used to extract rules, which were subsequently coded into templates. Therefore, in the final evaluation we will evaluate the extracted rules. But since the classifiers form the basis of the rules, they too have to be evaluated. In a first attempt to evaluate these classifiers, a 10-fold cross-validation was used with the different classifiers. The data was evaluated in a binary way, that is, a response produced by the classifier is matched against each suggestion of each user turn, and this either results in a correct or an incorrect answer. The results can be seen in Table 3.1. We used a decision tree classifier (Ripper) and a rule generating classifier (J48) to extract rules. Also, we used a SVM to see how a black box classifier would perform. In the results, we are mainly looking for a high precision, because the rules that suggest a certain response do not have to fire constantly, but if they do the response should be good.

Obadiah	Ripper P	Ripper R	J48 P	J48 R	SVM P	SVM R	Class
	0.47	0,69	0.52	0.54	0.56	0.67	Positive
	0.52	0.39	0.52	0.56	0.59	0.57	Sad
	0.17	0.04	0.75	0.13	0.80	0.17	General
	0.53	0.23	0.36	0.34	0.45	0.33	TellMeMore
	0	0	0	0	0.22	0.15	Laughter

Prudence	Ripper P	Ripper R	J48 P	J48 R	SVM P	SVM R	Class
	0.54	0.24	0.41	0.44	0.37	0.49	TellMeMore
	0.26	0.82	0.31	0.34	0.33	0.40	Sad
	0.25	0.02	0.33	0.27	0.33	0.24	Rational
	0.44	0.12	0.27	0.24	0.24	0.19	SadAngry
	0.65	0.31	0.50	0.51	0.57	0.48	NewSubject

Poppy	Ripper P	Ripper R	J48 P	J48 R	SVM P	SVM R	Class
	0.63	0.90	0.69	0.82	0.69	0.84	Happy
	0.26	0.10	0.32	0.19	0.39	0.24	NewSubject
	0.39	0.12	0.40	0.30	0.40	0.28	TellMeMore
	0.60	0.40	0.64	0.50	0.72	0.54	Sad

Spike	Ripper P	Ripper R	J48 P	J48 R	SVM P	SVM R	Class
	0.53	0.28	0.44	0.52	0.58	0.48	Insult
	0.58	0.37	0.53	0.52	0.48	0.47	TellMeMore
	0.36	0.85	0.40	0.51	0.34	0.49	Angry
	0	0	0.21	0.06	0.14	0.09	SadAngry
	0.60	0.26	0.56	0.48	0.64	0.46	Happy
	0.67	0.36	0.50	0.41	0.62	0.36	NewSubject

Table 3.1: shows the Precision and Recall of the different classifiers on the different response-groups.

3.2 Conclusions

The most important evaluation of the dialogue management will be done in the following months, during the user studies. During these months, data will be gathered to objectively evaluate the turn-taking, and an objective evaluation of the turn-taking rules is planned in these months too. During these months, especially the rules that were extracted from the classifiers will be evaluated.

However, the first evaluation of the classifiers shows that for a number of response-groups the classifier can fairly accurately predict when to say something from this group. For example, the classifiers for Poppy can predict a response from the 'happy' group (which contains responses if the user is acting happy) with an average precision of 0.66 and an average recall of 0.86.

4 License and availability

The dialogue components are available as part of the SEMAINE 3.0 system, under the LGPL license.

References

Bos, J.; Klein, E.; Lemon, O.; and Oka, T. 2003. DIPPER: *Description and formalisation of an information-state update dialogue system architecture*. In 4th SIGdial Workshop on Discourse and Dialogue, 115–124.

M. ter Maat, D. Heylen, *Selecting Appropriate Agent Responses based on Non-Content Features*, 3rd International Workshop on Affective Interaction in Natural Environments, at ACM Multimedia, Firenze, Italy, October 2010, To Appear

M. ter Maat, K.P. Truong, D.K.J. Heylen, *How turn-taking strategies influence users' impressions of an agent*. In: Proceedings of the International Conference on Intelligent Virtual Agents (IVA), 20-22 Sep 2010, Philadelphia, PA, USA. pp. 441-453. Lecture Notes in Computer Science 6356. Springer Verlag. ISBN 978-3-642-15891-9

M. Valstar, G. McKeown, R. Cowie, and M. Pantic. *The Semaine Corpus of Emotionally Coloured Character Interactions*. In Proc. IEEE Int. Conf. On Multimedia & Expo (ICME2010), Singapore, 2010. IEEE.

Appendix A – examples of rules

A1 Linking rules

This section contains some example rules of linking sentences.

Character	Previous agent utterance	User-turn features	Agent response
Poppy	Did things get better?	Disagree, short	Cheer up!
Poppy	There must be something that you're really looking forward to.	Disagree	Give me just one happy thought and you'll feel better.
Prudence	Would you say you have everything under control?	Agree, short	Good. You obviously have your head screwed on.
Prudence	Do you think that you are wise?	Agree, short	Really? Why?
Obadiah	Can you remember feeling more miserable?	Disagree, short	You don't need to pretend with me!
Obadiah	Don't you sometimes wish that you could just run away?	Disagree, short	Why? Life's a bitch and then you die.
Spike	Do you think anybody cares what you think?	Disagree	You've got rights too, you know.
Spike	And I suppose you're going to tell me nothing can possibly go wrong?	Agree	Well I think you're wrong to be so pragmatic about it all.