Dialogue Modeling in Embodied Communicational Agents

László Laufer
Department of Psychology, Budatest University of Technology and Economics
1111 Műegyetem rkp. 3-9, Budapest, Hungary
laufer@erg.bme.hu

Gábor Tatai
Department of Computer Science, University College London
Gower Street, WC1E 6BT, London, UK
g.tatai@cs.ucl.ac.uk

Bottyán Nemeth
AITIA Inc.
1117 Infopark st.1., Budapest, Hungary
bnemeth@aitia.ai

Abstract

We are developing an ECA (Embodied Communicational Agent) system in Hungarian language. The ECA has to be able to carry out general chat conversations and domain specific discussions as well. The structure of the dialogue graph and the attributes of the graph nodes have to support the success of the conversation: user utterance understanding and responding, as well as repairing the "non-understanding" situations. In the following sections we are briefly depicting the system and describe its dialogue knowledge base focusing on the different user interfaces we developed for its testing and building.

Keywords: Dialogue Modelling, ECA, Affective Computing

1. Introduction

The research on embodied conversational agents (ECAs) have intensified in the past few years as the demand on avatars representing users providing virtual presence over the network as well as autonomous chatterbots are widely applied in various applications by both the research community and industry. The most apparent application areas of ECAs are education, entertainment, search engines and general tools for information retrieval, virtual marketing, customer care and support etc.

It seems that the number of theoretical and practical implementation questions are growing exponentially as we are heading towards more and more complex applications. Some of these questions seem to be really tough both in theory and practice such as application of emotions, dialogue management, recovery of semantic information not mentioning the synchronization of all these. There are several papers addressing these issues and suggesting useful partial or global solutions [Cassell,Vilhjalmsson 1999]. In this paper we would like to give an overview of our ECA system, BotCom focusing on it's dialogue knowledge base. We also describe in detail the different tools we developed for editing and evaluating the knowledge base.

2. System overview

Our system, "BotCom" is an existing prototype system, part of a research project and also currently under introduction to industrial use. It is capable of chatting with users about different topics as well as displaying emotions corresponding to its written replies. The affective part of it's response is based on a complex emotional state generator, GALA.

Figure 1. Screenshot of an online tutoring agent explaining the definition of transaction in an E-learning application on financial markets. The user posed the question: “What does transaction mean?”, the agent gave a short answer, and displayed the section in the curricula explaining the asked definition.

Moreover, it has a feature of connecting to various information sources and search engines thus enabling an easily scalable knowledge base. The operation area and access medium of the BotCom is the Internet, therefore all constraints that apply to this medium had an effect to the possible implementation scenarios of the system. The primary use of BotCom will be interactive website navigation, entertainment, marketing and education.

2.1. Language

The language of the ECA is Hungarian, but with adequate language and semantic (WordNet) plug-ins the default language can be changed. There are several language dependent specialties that we will not discuss in details, however some major points are worth to be mentioned. Hungarian is an agglutinative language. The morphological rules are highly complicated. During the sentence processing we not only have to cope with hundreds of different appendices and their variations, but in some case the stem of the word changes as well. The different appendices added to the end of the words can change the meaning. The same words, with different punctuations can have different meanings, so we also have to take into account the mistyped words coming as user input and as a result, some misunderstandings are inevitable. Because of the above and the accented spelling of Hungarian, the language tools are highly complex. All these make it really challenging to create a ECA or even a simpler chatterbot in this language.

2.2. System Architecture

Before describing the dialogue editor tool and the dialogue knowledgebase of the chatterbot, let’s have a glance over the system architecture in order to understand better the functionalities of our ECA system, BotCom. It is primarily a client-server architecture with a separate dialogue analyzer and editor module.

The ECA is displayed in a web browser (see Fig. 1.). The client visualizes the messages of the chatbot, the animations representing its emotions and the gags, as well as forwards users’ messages towards the server.

The web pages where the ECA is embedded into are created and maintained by the Webra portal management tool (also our development, an independent Linux/ Apache/PHP based tool). Integrating the ECA into a portal engine is advantageous, because the knowledge base of the bot can be extended by expressions from the web site, since the portal engine stores the keywords and full articles in an indexed database. Thus the integration enables the access of this database.

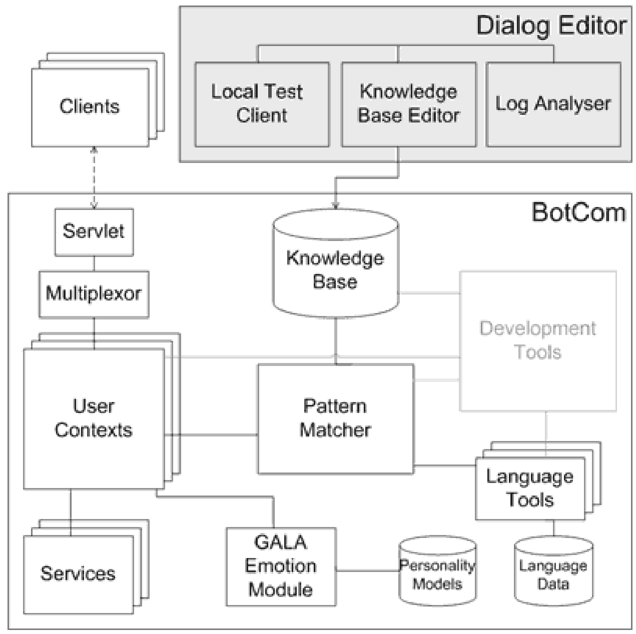

Figure 2. System architecture. The main components of BotCom system and the Dialogue Editor development tools.

At the server side a Java servlet collects the messages, transmits them to a multiplexer that queues or distributes them among the servers for parallel operation. Messages are processed separately, in the appropriate user context. Every logged in user has exactly one user context, which is responsible for the communication and stores all information about the user. The received user messages are handled by the dialogue system that matches them against the patterns stored in the knowledge base. If we find a matching pattern the response is being chosen, the answer which consists of the status obtained from the GALA emotional model and the browser controlling commands (if any) are sent back to the client afterwards.

2.3. The GALA Hierarchical Emotion Processing Model

In order to create a usable synthetic emotion model for a chatterbot that communicates mainly via text messages, it is essential to design an appropriate mapping scheme between the emotions and the expressions that the chatbot sends or receives.

Our goal was to select a sophisticated emotional model complex enough to provide ample workspace, but not bound to a net of cognitive processes. We have found that the model suggested by the psychologist Robert Plutchik to be an appropriate starting point for further description and generation of synthetic emotions [Plutchik 1980], [Plutchik 2001].
This model defines a cone as an emotion space and maps various emotions and intensities into it: in total, 24 emotions. This geometrical approach allows easy manipulation of the emotions. However, it has some drawbacks, as we have stated in a previous paper [Tatai et al 2003a], [Tatai et al 2003b] along with the detailed evaluation of the current version of the GALA engine.

In order to use a chatbot system equipped with emotional capabilities it is necessary to create a database that contains the mapping of words and expressions into emotions and more complex emotional structures. In our system users have the possibility of labelling parts of the text (words, expressions or entire sentences) with message acts or basic emotions, thus they can build up a complete expression-emotion mapping database. [Tatai, Laufer 2004]

3. Overview Of the Dialogue Knowledge Base

In the knowledge base the dialogue utterances are represented in a graph. Every message belongs to a topic and the topics are also arranged in a graph structure: they can be in a parent-child relation, if there is a sub-topic within one topic. They can also be in a neighboring relation: in case the robot decides to change the topic of discussion she goes on to a different, but related topic to address the partner. This proactive behavior is a key element of taking the dialogue initiative and shifting the focus of discussion to another topic, in which the chatbot has a bigger knowledge.

In human-computer dialogues the "non-understanding" situations occur more frequently than in everyday human conversations. Whereas in normal dialogues it happens at 5% of the cases, in chatter robot dialogues 20-25% is still deemed acceptable.[Schlangen 2004] One of the main reasons for this big difference is that misunderstandings up to a certain frequency are often a source of humor and therefore motivating for the users to get more engaged in the discussion.

In our experience the main problem of creating knowledge bases and dialogue engines for ECA's is detecting and repairing the potential misunderstandings in the dialogue. To this end, we are using a number of strategies. We created dialogue segments for *grounding*: utterances explicitly stating the focus of the utterance, explicit confirmation or rejection of the focus of the dialogue partner's utterance.[Paek, Horovitz 1999] For *re-identifying* or *re-grounding* the topic of the discussion we established separate parts in the dialogue graph. These segments can be topic specific, and general. After the semantic analysis of the user responses the system deducts the focus of the discussion and either repeats a modified version of the misunderstood message or jumps to the supposed topic. In some cases the robot can not shift the discussion back to an identified topic. In these cases she enters a neighboring topic, taking the *initiative* in the dialogue.

All these 3 functions can only be carried out successfully if the knowledge base contains sufficient dialogue segments, labeled with proper flags indicating their role for the dialogue engine.

3.1. Micro and Macro Level Management of the Knowledge Base

For a successful application the information in the knowledge base has to be sufficient on the micro and the macro level as well. Under the micro structure of the dialogue knowledge base we mean the consecutive user and robot utterances, and the attributes of each node in the graph. The macro structure is the overall usability of the graph, the completeness of the topics, the relation of the topics, and how the above listed 3 functions (*grounding, re-grounding* and *initiative*) are implemented in them.

3.2. Micro-management of the dialogue graph

For editing the micro level information in the knowledge base, the most important functions are: searching, establishing new relations between the nodes and editing their attributes. In our system a node can have three different types of properties.

The first property is the *topic* the node belongs to. As we previously described the topics are also forming a graph structure within the knowledge base.

The second type can be called *general node attributes*. These attributes assigned to a certain sentence describe:

- Whether the robot, the human dialogue partner or both can say this sentence.
- What are the variants of this utterance, which mean exactly the same, but are expressed differently?
- What kind of animations should be played while replying this sentence?

- Which URL should be displayed in the meantime?
- Whether the sentence can be used to initiate a discussion about it's topic?
- Whether it is a question, and if yes, what type?
- If the utterance is assigned to a certain robot personality or gender.
- The emotional load of the utterance, which we have already talked about in connection with the emotional subsystem. Both the user and the robot utterances can have an emotional load according to Plutchik's model of emotion.

The third type of node properties we call *dialogue variables*. These variables can be grouped in three categories:

- We use template sentences, where certain variables get their values from the user utterances. For e.g. if the bot wants to talk about the user's musical taste:
  Bot: What music do you like?
  User: I like $user_like_music$.
  Bot: I don't like $user_like_music$, I rather listen to bot_like_music.
  This way we get information about the user, and use it up in the generated sentences of the dialogue.
- Another type of variables are the general attributes of discussion. Whether the user is logged in or not, the chat has just been started or has been going on for some time. These variables influence the navigation in the dialogue graph, e.g. if the user is not logged in, the bot moves to the part of the knowledge base which encourages the user to do so.
- The emotional load of the user sentences can also be considered as a factor influencing the emotional state of the chatter robot in a certain way.

In order to display the dialogue graph both statically (Dialog editor window) and dynamically (Bot testing window) and to make it possible to set all these *node attributes* and *dialogue variables* for each utterance in the knowledge base, we created the following interface (see Fig 3.).

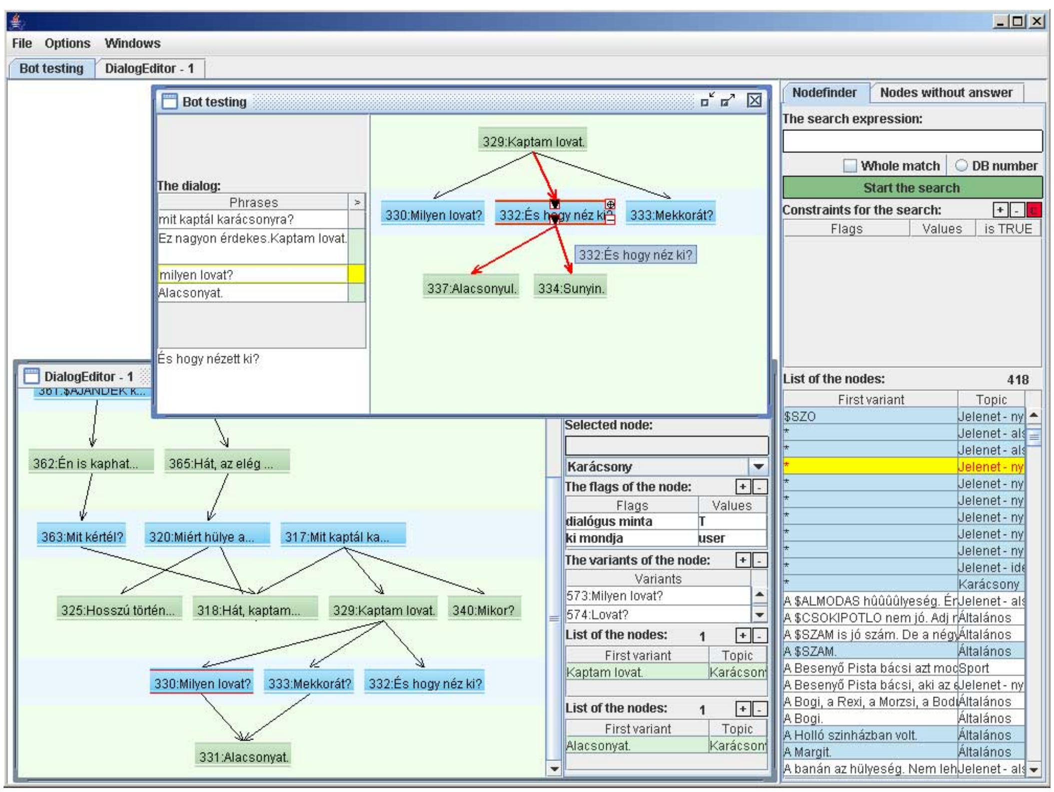

Figure 3. Dialogue editor. On the right side the node search panel is being displayed. On the left side a “Bot testing” window is open, where it is possible to test the chatter robot and follow the current and the other possible tracks of the conversation in the graph. Below a “Dialogue editor” window is open, where the dialogue graph can be edited directly.

On the left side of the main window is the search bar, enabling the user to search among the nodes by defining fragments of the utterance, or values of certain variables, or simply by topic. The selected node appears in a separate window, see the “DialogueEditor – 1” window in the lower left corner of Fig.3. This way several nodes and their surroundings can be seen at once. The active node is framed by lines on the top and the bottom, and its properties can be edited on the right hand side of its sub window (e.g. “DialogueEditor – 1”). The chatter-robot can also be tested through this interface, to see if the different weights of the possible responses are set properly, see “Bot testing” sub-window on Fig.3. As we proceed with the conversation the possible and the actual robot utterances and the identified user utterances appear below each other. In these displays one can always close and open the parents and the children of each node, and establish new connection between the nodes displayed in different sub-windows. For the visualization of the graph we are using an advanced form of the Dot’s algorithm of Gansner. [Gansner et.al 1993]

3.3. Macro-management of the dialogue graph

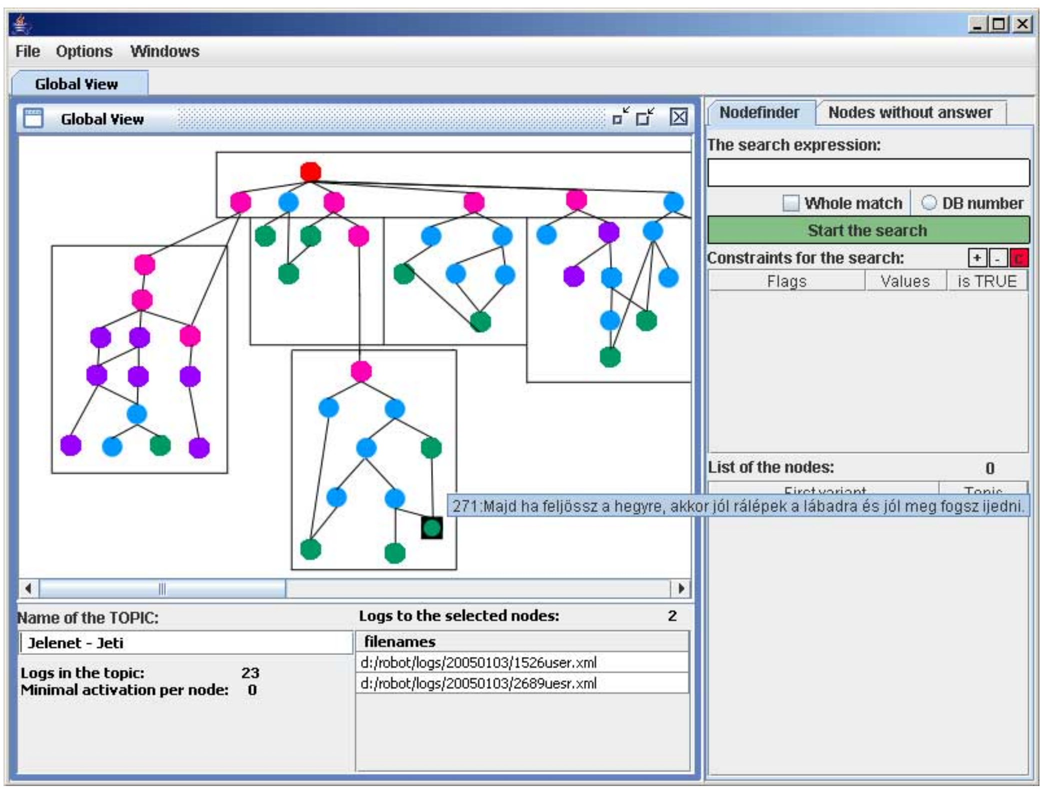

Figure 4. Global view. The dot with a black background is the selected node. In the bottom section of the window, the topic of the node and the dialogues (log files), where it was used are listed. In the upper parts we can get an overview of the knowledge base structure.

In the micro-management interface in the last paragraph we approached the dialogue from the level of the utterance. Therefore, the evaluation tool tested the conversation as a sequence in the graph. On the other hand, the macro-level evaluation of the dialogue graph uses the log files of the chatter-robot. After the selected log files are loaded (lower left corner of Fig.4.) the interface displays the nodes and their parents and neighbors grouped by subtopics. For graph drawing we use a method similar to the methods of one-dimensional optimization by L. Carmel et al. [Carmel et al. 2001]. The different colors of the nodes represent the frequency of their appearance in the log file. After selecting a certain node in the graph one can read the exact utterance it represents and also the log files it appears in. The figure also shows that as we go lower in the structure the nodes become less frequent (i.e., they change from purple, through dark blue, to light blue and to green).

This macro management tool was designed to see how the areas of the knowledge base contribute to the actual dialogues carried out by the users of the chatter robot in the past. The interface can disclose whether a part of a topic or subtopic is rarely or never used, and thus should be modified in order to contribute to the intelligence of the robot. On the other hand it can also show the most popular user paths in the graph, which need further development to enrich the variety of responses of the robot, and to extend the range of understood user sentences.

4. Summary

Creating an ECA system involves a big variety of developments. After establishing the system architecture and the dialogue engine, one has to integrate it into a multimodal user interface. It is also important to detect and to react on the users' emotions adequately to facilitate user engagement. Still one of the most important elements in creating a successful ECA system is the dialogue modeling and representation subsystem. Because of the practical nature of the application, the starting point of editor development is the usability evaluation of the knowledge base. However, the evaluation criteria are different at the micro and at the macro level. In the first case, the conversation is tested by investigating the possible conversations; in the second case the dialogues carried out in the past are investigated. Both cases the grounding, re-grounding and initiation are the capabilities that have to be examined and improved. It is also important to create a user interface, where the general node attributes and the dialogue variables of an utterance can be set thus enabling them to fulfill their proper role in the conversation.

5. References

L. Carmel, D. Harel and Y. Koren: "Drawing Directed Graphs Using One-Dimensional Optimization ", Technical Report MCS02-14, The Weizmann Institute of Science (2001)

Justine Cassell, Hannes Vilhjalmsson: Fully embodied conversational avatars: Making communicative behaviors autonomous. Autonomous Agents and Multi-Agent Systems 2: 45-64. (1999)

E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo: "A Technique for Drawing Directed Graphs", IEEE-TSE 19:3 (1993)

Michael A. Gilbert: "Prolegomenon to a pragmatics of emotion", Proceedings of the Ontario Society for the Study of Argumentation. Brock Univer-sity, St Catherine's, Ontario, Canada (1997)

Tim Paek, Eric Horvitz: "Uncertainty, Utility, and Misunderstanding: A Decision-Theoretic Perspective on Grounding in Conversational Systems", Working Papers of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems (1999)

Ana Paiva: "Affective interactions: Toward a new generation of computer interfaces?", Springer-Verlag Berlin and Heidelberg GmbH & Co. (2000)

R. Plutchik: A general psychoevolutionary theory of emotion. Plutchik, R. and Kellerman, H. (eds.): Emotion theory, research, and experience. 3-33 pp. Academic Press. (1980)

R. Plutchik: The nature of emotions. American Scientist 89(4): 344-350. (2001)

G. Tatai, A. Csordás, Á. Kiss, A. Szaló, L. Laufer: “Happy chatbot, happy user”, Proceedings of the 4th International Working Conference on Intelligent Virtual Agents (IVA'03), Irsee, Germany, Springer Verlag (2003a)

G. Tatai, A. Csordás, Á. Kiss, L. Laufer, A. Szaló: The chatbot who loved me, *Proceedings of the AAMAS 2003 workshop on Embodied Conversational Characters as Individuals*, Melbourne, Australia (2003b)

G. Tatai, L. Laufer: Extraction of Affective Components from Texts and Their Use in Natural Language Dialogue Systems, Acta Cybernetica 16, 625-642, Szeged, Hungary (2004)

David Schlangen: “Modelling and Managing Dialogue, Approaches and Challenges” (2004)