Implementing a Multi-user Tour Guide System with an Embodied Conversational Agent

Aleksandra Čereković[1], Hsuan-Hung Huang[2], Takuya Furukawa[2], Yuji Yamaoka[3],
Igor S. Pandžić[1], Toyoaki Nishida[2], and Yukiko Nakano[3]

[1] Faculty of Electrical Engineering and Computing, Zagreb, Croatia
[2] Graduate School Informatics, Kyoto University, Japan
[3] Dept. of Computer and Information Science, Faculty of Science and
Technology - Seikei University, Tokyo, Japan
{aleksandra.cerekovic,igor.pandzic}@fer.hr,
{huang,furukawa,nishida}@ii.ist.i.kyoto-u.ac.jp,
{50007646208,y.nakano}@st.seikei.ac.jp

Abstract. In recent years, computer-generated interactive virtual characters, called Embodied Conversational Agents (ECAs), are subjects of considerable ongoing research. Nevertheless, their conversational abilities are mediocre compared to real human behaviors. Among limitations, most of ECAs are incapable of participating in natural conversations in which the number of participants can change dynamically. In the ongoing work we investigate principles of integrating a multi-user support in an ECA system. We present experiments and implementation approach of a prototype system in which a tour guide ECA interacts with one or two users. The system combines different technologies to detect and address the system users and draw their attention. Experimental interaction with the system produces encouraging results. The system can address the user's appearance, departure, decreased level of interest and identify his conversational role.

1 Introduction

In recent years, computer-generated interactive virtual characters, called Embodied Conversation Agents (ECAs) [14], are subjects of considerable ongoing research. ECAs have potential to behave like humans, and thus an opportunity to achieve naturalness in human-machine interaction. It has been proven that the presence of ECAs can improve a human's engagement and significantly increase his positive perception of learning experiences [10].

Nevertheless, conversational abilities of present-day ECAs are mediocre compared to real human behaviors. Among other limitations, most ECAs are incapable of participating in natural conversations in which the number of participants can change dynamically. In the theory of multi-party human-agent communication there are several bases which discuss multi-party dialogue issues and human's nonverbal conversational signals during interaction. Still, multi-user ECA systems are experimental and scarce. Nowadays technology is too immature to overcome challenges of bringing multi-party support to human-agent conversational systems; e.g. requirements to track

J. Liu et al. (Eds.): AMT 2009, LNCS 5820, pp. 7–18, 2009.
© Springer-Verlag Berlin Heidelberg 2009

and update user state, detect and resolve their requests in real-time, handle complex conversations…

In the ongoing work we investigate principles of integrating multi-user support in an ECA system. We present implementation approach of a prototype system in which a tour guide ECA interacts with one or two users. The system solves some of the multi-party dialogue issues presented in the next section. It can detect a user's arrival, departure, decreased level of interest and identify his conversational role. It can also recover from system failures, e.g. failure of speech recognition.

The paper is organized as follows. First, we present related work and discuss key challenges and the conversation model. The system design and architecture are explained in the fourth section. We conclude the paper with a summary of future research directions and brief discussion.

2 Related Work

In the theory of multi-party dialogue interaction between agents and humans, most of the work has been done by Traum [1] [9] [12]. For the purposes of the Mission Rehearsal Exercise (MRE) Project [12], Traum investigated multi-party dialogue issues and grouped them into three parts [9]:

- **Participants' roles.** The issue refers to identification of participants' local roles and responsibilities which shift during interaction (who is addressee, listener, and speaker). In addition to that, it also refers to participant's social roles and their effect on interaction behaviors (e.g. status, relationship).
- **Interaction management.** Managing the communication flow in a multi-party system is far more difficult than in a dyadic system. Some of the difficult issues are how to give and recognize a turn and how to handle participants' channels (and backchannels). Besides, conversations can be easily split and merged together and attention can be paid to several persons.
- **Grounding and obligations.** are notions commonly used to model local state of dialogue. In multi-party communication usage of these models can become very complex; e.g. if there are more addressees, it can become unclear what a proper grounding should be.

Traum's paper only discusses issues which arise in multi-party human-agent systems, but does not give practical solutions to it.

Vertegaal et al. [7] focus on human gaze behaviors in multi-party conversation. They studied a real three-party conversation and discovered that the subjects look about seven times more at the individual they listen to than at others, and about three times more at an individual they speak to than at others. The conclusion is that gaze is an excellent predictor of conversational attention in multi-party conversations.

Similar to the functionality of our system, Rehm and Wissner [8] developed a gambling system in which an ECA plays a dice game with two humans. As game rules define conversational turns, their system has a simple dialogue model in which turn-taking mechanism and participants' roles are round-based. The system lacks the active gaze model which follows human users. In further study, Rehm and Andre [11] investigate human gazing patterns in interaction with a real human and an agent. They note that people spend more time looking at an agent that is addressing them than at a human speaker. This phenomenon can be explained by the fact that prolonged eye contact in a social interaction can be considered impolite and rude; hence, the agent in this game may have been regarded as an artifact rather than a human being.

3 Key Features and Initial Experiments

In addition to the findings from multi-party dialogue theory, at the very beginning of development we identified key features which a multi-user ECA system should possess. Afterwards, we proceeded with several experiments with the system equipment to determine its final configuration, which is shown in Figure 1.

- **Speech processing.** If the system users are standing next to each other the microphones should be localized to avoid an overlap between their verbal channels. Conducted experiments showed that Loquendo ASR [18] meets our needs very adequately; it is stable in noisy environments, speaker-independent, there is no significant speech overlap at reasonable distance (0,5 meters approx.), and has keyword spotting feature absent from the free Microsoft SAPI engine. However, Loquendo sometimes reacts to voices which are not from the users. We think such errors can be prevented by using the Speaker Verification function, which we have not yet tested.
- **Nonverbal data acquisition.** During interaction it is extremely important to detect and locate the users so the ECA can gaze at them. For that purpose we installed two cameras and used image processing techniques to detect the user's arrival, position and departure.

  To process image from one camera we decided to use Okao library [16] which provides accurate face detection and extra features such as face orientation, gaze direction, the positions and openness of eyes and mouth, gender detection, age identification and face identification from a single image. In preliminary tests with a 960x720@15fps web cam, accuracy of face detection was sufficiently high and undoubtedly usable, but most of the other functions were not reliable and could only be treated as an extra bonus. Since Okao Vision does not require stereo cameras, this result was acceptable for our system.

  The second camera recognizes moving objects and detects the user's arrival and departure by comparing the differences between sequential images coming from the camera's input.
- **The conversation model.** The final conversation model we designed simplifies participants' conversational roles and is based on narration. In the system scenario, an ECA named Dubravka takes visitors on a tour of the city of Dubrovnik and talks about its cultural heritage, history, and monuments. She maintains the initiative in the conversation, though users can interrupt her and ask questions about the current topic. We predict topic-related questions and define it by using specific keywords in the speech recognition engine (such as "where, "when", "how"). To hold the users' attention during the session, Dubravka also asks the users simple "yes/no" questions.
- **ECA appearance.** The users should perceive the ECA's gaze direction, which has proven to be very important in regulation of conversation flow in multi-party

communication. Since the system environment is displayed on a 2D screen, the ECA’s size and an appropriate gaze model are important to avoid the Mona Lisa effect (an impression perceived with 2D pictures of humans, that “the eyes follow the observer across the room”). The ECA's final position on the screen was derived from initial experiments with the ECA’s size and its attention towards the users. We concluded that users can distinguish the direction in which the ECA is gazing only if the ECA's size on the screen is very large.

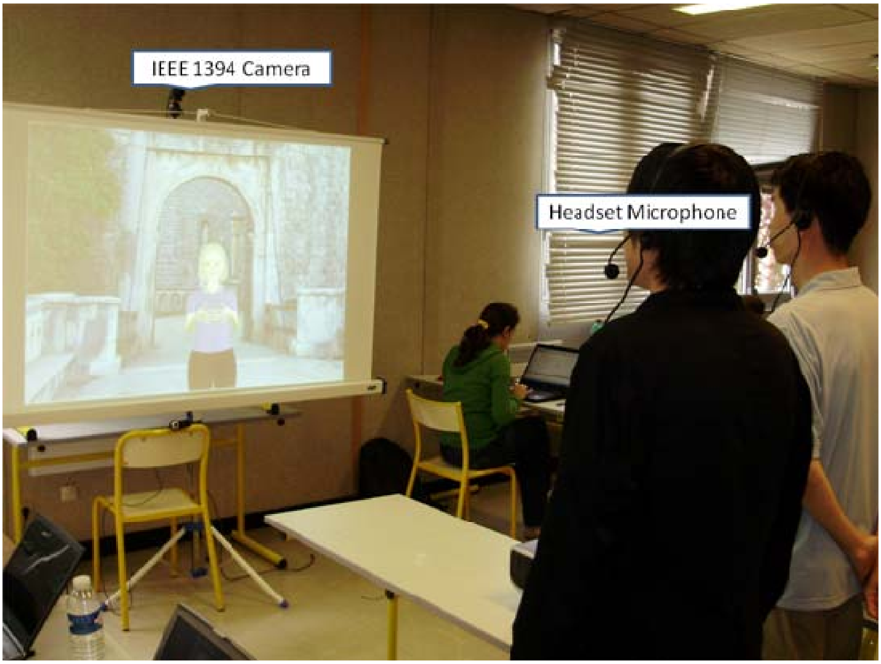

Fig. 1. Final system configuration

4 The Conversation Model

4.1 Multi-party Features

The conversation model is designed to handle situations typical for multi-party interaction. When designing the model we predicted specific situations, such as speech collision and mutual conversation between the users, and defined the system’s response. We discuss multi-party features of the conversation model and system with regard to issues identified in Traum's paper [9]:

- **The user’s appearance.** During interaction, the system can dynamically detect the user’s arrival, his position in front of the system and his departure. In its initial state, Dubravka waits for someone to appear. When that happens, she approaches and engages the potential user. In case that user departs, she turns back and reverts to the initial state. At most two users can stand in front of the system. If there are less than two users, Dubravka can also invite another person to join the conversation.

- **Channel management.** The system combines users' verbal and nonverbal channels to resolve their utterances. Nonverbal behaviors taken into account are face orientation and gazing. By combining those modalities the system is able to define one of the following situations: decreased level of attention, making requests of the system, departure, speech collision.
- **Speech collision.** During interaction it might happen that the users ask questions simultaneously. The system handles this situation by having the ECA address one of the speakers and give him turn, e.g. Dubravka says "You can speak one by one (gesture "calm down"), I'll respond to you both. You can be the first one (addressing one user by pointing with an open hand)"
- **Identification of conversational roles.** Our conversation model simplifies the rules of how participants' local roles shift during the session. System scenario defines the communication workflow and several specific situations:
  1. When talking about cultural heritage of the city, Dubravka gazes at both users with the same frequency. In the case when one of the users asks a question, the system identifies him by using a localized microphone, and when Dubravka responds to him, he becomes an addressee, and the other user becomes an overhearer. Following the findings from studies on gazing [7, 11] we developed a computational model of gaze in which the agent gazes at the addressee more than he does at the overhearer.
  2. During the session, Dubravka may ask the users simple questions, e.g. "Is this your first time in Dubrovnik?" In this case she waits for both reactions and responds to each user separately. The reason why we added a questionnaire into the story is to hold the users' interest in the session.
  3. As described, most of the time the agent is the speaker and one or both users are addressees. Unfortunately, natural conversation between three parties in which each individual can become speaker, listener or addressee is not handled in a natural way. Since our agent is capable of understanding only simple questions related to the current topic, conversation between users is not welcome.

4.2 Additional Features

We are also concerned with features which can make interaction with the system more fluid. It is important to recover the system from failures, such as failure of speech recognition. In the case when a user's speech is not recognized, we propose two-stage recovery. First Dubravka asks the user to repeat his question. If his speech is not identified the second time around, she responds to him: "I'm sorry. I don't know a response to your question".

We also use Okao vision to recognize a situation in which the user is paying less attention to the session. If this occurs, Dubravka speaks until the end of the planned utterance, turns to the user and says "Seems you are not interested in this topic. Would you like to hear something else?"

5 System Overview

The system is constructed from several computers and hardware equipment which communicate through TCP/IP-based OpenAir routing protocol [20]. Implementation of the OpenAIR server and plugs is used to set up the GECA Framework, a software framework which enables rapid integration of different ECA modules. Apart from the server and plugs, GECA framework also defines communication rules between ECA components. More details about GECA can be found in the work [5].

5.1 System Design

System design is depicted in Figure 2, which depicts the software components connected to the GECA platform/server. The components communicate through the server using GECA messages. As part of the message content we introduce a variable system which describes interaction between the agent and humans. For example, `SpeechInput` represents the most recent result from one of the speech recognition components, `Speaker` represents the id of the speech recognition component, `UserNumber` represents the number of users who are standing in the user area, `UserStatus` represents the availability of the users, `UserAttention` represents how much the users are paying attention to the system, `Addressee` specifies the addressee of the agent's next utterance, etc. During the session, depending on the situation in the environment, the system components update the values of variables and exchange data to define a final response.

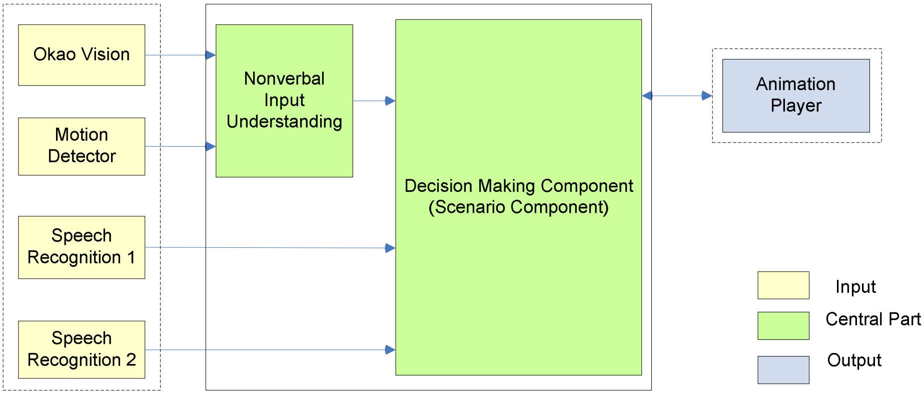

Fig. 2. Design of the system components. Blue arrows represent message workflow.

5.2 System Components

The system components can be roughly divided into three categories: input, central part, and system output.

System Input
Input components detect and process human speech, appearance and face orientation. Combination of this data maps to one of several distinct situations (Table 1) in the system environment.

Table 1. Situations in the system environment and explanations how each situation is detected

Situation/Feature	Situation Description	Components which detect situation	Explanation how components work
User's arrival	System is in idle state until it detects potential users	Motion Detector	Motion Detector detects movements in the area which activate the system.
Number of users	During interaction system tracks the number of users	Okao Vision & Motion Detector	Okao vision detects number of faces. If it fails, Motion Detector processes users' areas on the image and detects motions in each area.
Interruptions	User interrupts Dubravka to make a request	Speech Recognition	For each utterance we predict topic-related questions defined with keywords as "how", "where"…
Unrecognized speech	Subtype of interruption; request is not recognized	Speech Recognition	Speech is detected, but no keywords are triggered. Loquendo SAPI has additional features for this purpose
Decreased level of attention	Users are not paying attention to the system, e.g. they gazing around	Okao Vision & Motion Detector	Okao Vision processes user's facial orientation and Motion Detector handles their movements.
Leaving the system	User(s) is(are) leaving the system	Speech Recognition, Okao Vision, Motion Detector	User can say "bye, bye" or "I'm leaving" which triggers speech recognition component. If there are no motions or faces on the input images, users have departed.

Speech recognition. Component is based on Loquendo ASR [18]. Once the component is started, it awaits users' speech input. When it receives a speech start signal, it starts recognizing the speech. The results of recognition, timing and speaker's id, are then sent to the server and passed to Decision Making Component.

Nonverbal Behaviors. To detect user's nonverbal behaviors we developed two input components: Okao Vision and Motion Detector.

- **Okao Vision** component is based on Okao Vision library [16]. The system uses the following features of Okao: face detection, face orientation tracking and eye-mouth openness. Face orientation is used to approximately determine the users' gaze direction. This is not very accurate but should be sufficient when we only need to distinguish rough directions like the screen or another user. Results of the component are sent to Nonverbal Input Understanding component through server.
- **Motion Detector**, the component which we use to detect moving objects in distinct areas, is based on OpenCV library [17]. The component divides the viewed image region coming from the camera into two distinct areas (which we refer to as *user areas*) and detects motion inside each area. The user areas are surrounded with a blue and a red square, respectively. To recognize moving objects we calculate a

sum of pixels in one frame and compare this value to the value from the previous frame. If threshold is exceeded, moving objects are detected.

Central Part of the System

Nonverbal Input Understanding component uses information received from users' nonverbal channels to detect some of the situations defined in the system scenario.

First, it combines input data coming from Motion Detector and Okao Vision and uses simple heuristic methods to resolve the number of users. Okao Vision fails in detection when users rotate their head beyond 60 degrees, so it is not sufficient to track the number of users. However, we can determine the user's presence by detecting movement in the user area with Motion Detector. For example, during the system session two users listen to the agent and the left user turns his head to see who entered the room behind him. In this situation Okao Vision sets `UserNumber` variable to one, as if there is just the right user in the system, and sends it to the Understanding component. At the same time, Motion Detector detects motions in the left area and notifies the Understanding component. It then sets `UserNumber` value to two and sends it to the Decision Making Component.

Nonverbal input understanding component also tracks users' attention. We identify two meaningful patterns the users tend to look at for a significant proportion of time – the agent and the other user. Okao Vision is used to detect face orientation which we assume is the user's gaze direction. Since two patterns (agent, other user) are placed in different directions this approach is satisfying to efficiently track the user's attention. By combining results of Okao Vision and Motion Detector, the system can smoothly respond to situations in which a user is not interested in the system anymore.

Decision Making Component. This component is implemented based on Information State Theory [2] [3], which is a general theory of human-agent dialogues. A dialogue based upon this theory is defined as a set of variables (or information) that describe the current state of the dialogue. The implemented prototype is based on information state dialogue move engine [4] and capable of handling multi-modal, multi-party conversations, dynamically changing behaviors accompanying the emotion dynamics simulating component, etc.

To support the concepts proposed in the theory, we developed a script language [5] based on AIML [22]. In AIML, possible human-agent interactions are defined as one-to-one pairs. Compared to AIML which merely matches recognized speech inputs and non-verbal inputs with predefined patterns, we introduce a variable system to describe interaction between the agent and humans. Values of these variables are updated with the agent system's internal status and perception events sent from the speech recognition and nonverbal input interpretation components. Script designers can also specify variable update logic as effects of particular input patterns. `Effect` element is introduced into the `Template` element for this purpose. An input event can cause the values of particular variables to be bound to, added with, or subtracted from certain values.

The pattern syntax is also extended. Predicate element is introduced to represent a test of variable values. It is possible to test if the value of a variable is equal to, lesser or greater than another value.

The chatbot-like ECA system is then extended to a more powerful rule-based autonomous system. The agent or the script execution engine updates its internal status variables based on perception of the outside world or users, and picks for execution the first valid template for which all conditions (predicates) evaluate as true. Therefore, rules such as the one that specifies what the agent should do when a user appears in or departs from a user area can be specified in scripts.

States limit possible patterns that will be used in matching in the current conversational situation and thus isolate interference from other states which may happen to have the same triggering patterns. Due to the absence of a context management mechanism in the agent's behavior control, there is no way to determine whether a user's answer is related to the last question asked by the agent. However, for example, when the agent is going to ask a yes/no question such as "Do you need a tour guide?", transition to a specific state corresponding to the question can isolate it from other yes/no questions.

`GlobalState` is introduced for error and interruption handling. When a failed or unknown recognition occurs, appropriate response will be sought among the categories defined in the global state. When interruptions from the user such as "excuse me" or "pardon" occur, they are also matched against the patterns defined in this state.

The disadvantage of this approach is that, in absence of a full dialogue-managing central component, the agent does not conduct a plan that contains multiple steps to achieve a certain goal. The agent's behaviors are driven by the events that occurred in the outside world. The management mechanism for information like grounding or topics is not included in the script execution kernel.

In this version of the system which is still a work in progress, only a few issues specific to multi-party conversation are addressed. The gaze direction of the agent, which is essential in three-party dialogue, is not controlled by the central component but by the player. Control is done by using the `Addressee` attribute introduced in the `Utterance` element to specify if the addressee of an utterance is located to the left, right or both. Addressee specification is done by the rules in the scenario script by the script programmer; e.g. a specific utterance of the agent can be directed at the last speaker, and so on.

System Output

Animation Player. The role of the Animation Player is to produce a final output which represents the response to situations in the system environment. It displays the city of Dubrovnik where the tour guide ECA Dubravka stands and communicates with the users. During the session she walks through the city, shows them landmarks and reacts to their interruptions, questions and departures.

The animation system is built to the following requirements:

- **Virtual environment for the ECA.** The Player is based on visageISDK, an MPEG-4 compatible character animation framework [15]. As this product does not have support for virtual environments, the player uses 2D images of the city of Dubrovnik as background. Position of the agent is calculated using ARToolkit software, which tracks a marker to position the ECA [6]. Final output gives an impression of the agent standing within the city.

- **Ability to run multimodal behaviors described with GSML syntax.** During the session Animation Player receives messages from the Decision Making component. Messages contain GSML description of the ECA's multimodal behavior which should be run in real time. For that purpose we developed a thread-based mechanism which schedules and synchronizes running nonverbal behaviors with speech. The agent's nonverbal behaviors are produced and tested with GSML syntax before the system is integrated. Animations for the agent are either modeled manually in 3ds Max or procedurally in Visage. At the moment the system uses 25 modeled hand gestures in total (points, beats, symbolic). Procedural animations are head movements (nod, shake, tilt), facial expressions (joy, anger, fear, surprise, sadness), gazing and walking.
- **Responding to interruptions and departures.** The Animation Player should have smart scheduling so it can immediately respond to situations from the environment, even if it is running behaviors for the agent. We created a simple scheduler which immediately stops running behaviors when the Decision Making component sends a `Stop` message and resets the Player to initial state when users leave.
- **Gaze at the user.** Gazing model runs on the utterance level and is controlled by the `Addressee` attribute. E.g. if the addressee is the left user, the agent will gaze first at the left side, and then it will glance to the right for a while and gaze back to the left user again. Since we cannot predict the duration of the utterance, we repeat the patterns and stop when the utterance is finished.

6 Conclusions and Future Work

In this paper we presented an implementation approach of a prototype system in which an Embodied Conversational Agent communicates with two human users. Although no system evaluation has yet been performed, the initial results are encouraging. The system can correctly detect the user's arrival and departure, distinguish local roles of conversation participants and use ECA's gazing direction to specify an addressee. It can react to interruptions such as user requests and it can shut down automatically after users' departure. The implemented features execute smoothly, except when the user is leaving, when a certain delay is observed.

As future directions we plan to address two implementation additions which should make the interaction with the system more fluent. Limitations of Speech Recognition component make the agent capable of understanding only topic-related questions. As the agent cannot process discussion between the users, we aim to detect this situation and handle it in a polite way. If this happens, the agent will try to draw attention and continue narration about current topic. For example: "(waving) Alright, let's continue with the tour." Furthermore, we aim to extend the system to dynamically invite observers to join the session, which should make the interaction interesting. Example is the situation when one user leaves and there is one observer standing in the background. In this situation, the agent will look at the observer and say "We have one free place, please come closer and join the tour". This feature demands more complex image processing than what we currently perform on our camera input. After we implement these features we intend to perform system evaluation. In particular we are interested in the level of the interaction naturalness and how it can be increased.

In parallel, to make the system components open to the ECA community, we are upgrading the Animation Player with support for Behavior Markup Language (BML) [13][21][20]. BML is a language for describing physical realizations of multimodal human behaviors and it appears to have been well-received by the research community.

The weakest point in the system is the rather simple conversation model which limits the ECA's abilities to respond to questions we did not take into account. Nevertheless, we find this work useful for further studies and experiments in the field of multi-party interaction between ECAs and humans.

Acknowledgement

This research is partially supported by the Ministry of Education, Science, Sports and Culture of Japan, Grantin-Aid for Scientific Research (S), 19100001, 2007,"Studies on Construction and Utilization of a Common Platform for Embodied Conversational Agent Research" and the Ministry of Science Education and Sports of the Republic of Croatia, grant nr. 036-0362027-2028" Embodied Conversational Agents for Services in Networked and Mobile Environments." We would also like to thank Dr. Lao Shihong from OMRON Corporation Research & Development Headquarters for licensing the OKAO Vision.

References

[1] Traum, D., Rickel, J.: Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds. In: AAMAS 2002, vol. 2

[2] Traum, D., Bos, J., Cooper, R., Larsson, S., Lewin, I., Matheson, C., Poesio, M.: A model of dialogue moves and information state revision (1999)

[3] Larsson, S., Berman, A., Gronqvist, L., Kronlid, F.: TRINDIKIT 3.0 Manual. Trindi Deliverable D6.4 (2002)

[4] The MITRE Corporation: Midiki User's Manual, version 0.1.3 beta edition (2005)

[5] Huang, H., Cerekovic, A., Pandzic, I., Nakano, Y., Nishida, T.: The Design of a Generic Framework for Integrating ECA Components. In: Proceedings of 7th International Conference of Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 2008, pp. 128–135 (2008)

[6] Huang, H., Cerekovic, A., Tarasenko, K., Levacic, V., Zoric, G., Pandzic, I., Nakano, Y., Nishida, T.: An Agent Based Multicultural Tour Guide System with Nonverbal User Interface. The International Journal on Multimodal Interfaces 1(1), 41–48 (2007)

[7] Vertegaal, R., Slagter, R., van der Veer, G., Nijholt, A.: Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2001, Seattle, Washington, United States, pp. 301–308. ACM, New York (2001)

[8] Rehm, M., André, E., Wissner, M.: Gamble v2.0: social interactions with multiple users. In: Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, The Netherlands, July 25 - 29 (2005)

[9] Traum, D.: Issues in multiparty dialogues. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 201–211. Springer, Heidelberg (2004)

[10] Lester, J.C., Converse, S.A.: The Persona Effect: Affective Impact of Animated Pedagogical Agents. In: Pemberton, S. (ed.) Human Factors in Computing Systems: CHI 1997 Conference Proceedings, pp. 359–366. ACM Press, New York (1997)

[11] Rehm, M., Andre, E.: Where do they look? Gaze Behaviors of Multiple Users Interacting with an Embodied Conversational Agent. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 241–252. Springer, Heidelberg (2005)

[12] Traum, D., Marsella, S., Gratch, J., Lee, J., HartHolt, A.: Multi-party, Multi-issue, Multi-strategy Negotiation for Multi-Modal Virtual Agents. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 117–130. Springer, Heidelberg (2008)

[13] Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Thórisson, K., Vilhjálmsson, H.: Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 205–217. Springer, Heidelberg (2006)

[14] Cassell, J.: Embodied Conversational Agents. MIT Press, Cambridge (2000)

[15] http://www.visagetechnologies.com

[16] http://www.omron.com/r_d/coretech/vision/okao.html

[17] http://sourceforge.net/projects/opencvlibrary/

[18] http://www.loquendo.com/en/

[19] http://en.wikipedia.org/wiki/OpenAIR

[20] http://wiki.mindmakers.org/projects:bml:main

[21] Vilhjálmsson, H.H., Cantelmo, N., Cassell, J., Chafai, N.E., Kipp, M., Kopp, S., Mancini, M., Marsella, S.C., Marshall, A.N., Pelachaud, C., Ruttkay, Z., Thórisson, K.R., van Welbergen, H., van der Werf, R.J.: The behavior markup language: Recent developments and challenges. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 99–111. Springer, Heidelberg (2007)

[22] A.L.I.C.E. AI Foundation: AIML, Artificial Intelligence Markup Language (2005), http://www.alicebot.org/TR/2005/WD-AIML