Published in Proceedings of the 9th International Conference on Telecommunications ConTEL 2007

Towards an Embodied Conversational Agent Talking in Croatian

Aleksandra Čereković*, Hung- Hsuan Huang**, Goranka Zorić*, Kateryna Tarasenko**, Vjekoslav Levačić*, Igor S. Pandžić*, Yukiko Nakano*** and Toyoaki Nishida**
* Faculty of Electrical Engineering and Computing, Zagreb, Croatia
** Graduate School Informatics, Kyoto University, Japan
*** Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture & Technology, Japan
(aleksandra.cerekovic, goranka.zoric, vjekoslav.levacic, igor.pandzic)@fer.hr
(huang, tarasenko, nishida)@ ii.ist.i.kyoto-u.ac.jp
nakano@cc.tuat.ac.jp

***Abstract*—The advancement of traffic makes world more and more internationalized and increases frequency of communication between people who come from different cultures. Differences in their conversation go beyond the languages they speak to the non-verbal behaviors they express while talking. To improve the abilities of the embodied conversational agents (ECAs) while interacting with human users we are working on the ECA based application, a tour guide of city Dubrovnik that servers visitors in Japanese, Croatian and general western cultures speaking in English. This paper presents the overall architecture and explains possible extension with another culture. It describes issues we met while making agent talk in Croatian and proposed solutions.**

Index Terms— embodied conversational agent, distributed system, blackboard, user interface, Croatian speech recognition, Croatian speech synthesis

I. Introduction

Embodied Conversational Agents (ECAs) are computer generated human-like characters that interact with human users in face-to-face conversations [1]. ECA based systems express modalities natural to human conversation: speech, facial and hand gestures, emotions, and body postures. To achieve these modalities, system has to detect and recognize user's verbal and non-verbal input, and according to the state of conversation, respond to it in a real time. These features involve many research disciplines such as: natural signal processing, sensor signal processing, verbal and non-verbal behavior understanding, facial expression recognition, dialogue management, personality modeling, emotional modeling, natural language generation, facial expression generation and gesture generation.

A certain number of the ECA based architectures is previously proposed, but none of them has a general use. Moreover, components of individual research do not have interoperability among the various systems. Therefore, our research group is developing a generic ECA backbone framework, GECA framework [2] that connects a set of modulated ECA software components. The purpose of our work is to provide a rapid building of the multimodal interfaces and to prevent the redundant efforts and resource uses. To perform evaluation of the GECA framework, we applied to it several test applications. In this paper, we are going to present the system architecture of the first serious application, important issues and additional efforts we did to make our goals.

The advancement of traffic makes world more and more internationalized and increases frequency of communication between people who come from different cultures. Differences in their conversation go beyond the languages they speak to the non-verbal behaviors they express while talking. To improve the abilities of the embodied conversational agents (ECAs) while interacting with human users we are working on the ECA based application, a tour guide of city Dubrovnik that servers visitors in Japanese, Croatian and English. A tour guide system is proposed as a project under the title "An Agent Based Multicultural User Interface in a Customer Service Application" during the period of the eNTERFACE '06 workshop [1]. eNTERFACE '06 belongs to series of workshops on the multimodal interfaces and was held during the four-week period of the summer 2006 in the city of Dubrovnik. The aim of eNTERFACE workshops is to establish a tradition of collaborative, localized research and development work by gathering a group of researchers during the one-month period. Our project had involved 6 students and 3 senior leaders. During the relatively short-term period we had developed an ECA based tour guide application of the venue of the workshop, Dubrovnik city, specified as a UNESCO Worlds Heritage (Figure 1). After the workshop ended, we continued to improve a tour guide system.

Figure 1. Interface of the ECA based tour guide system

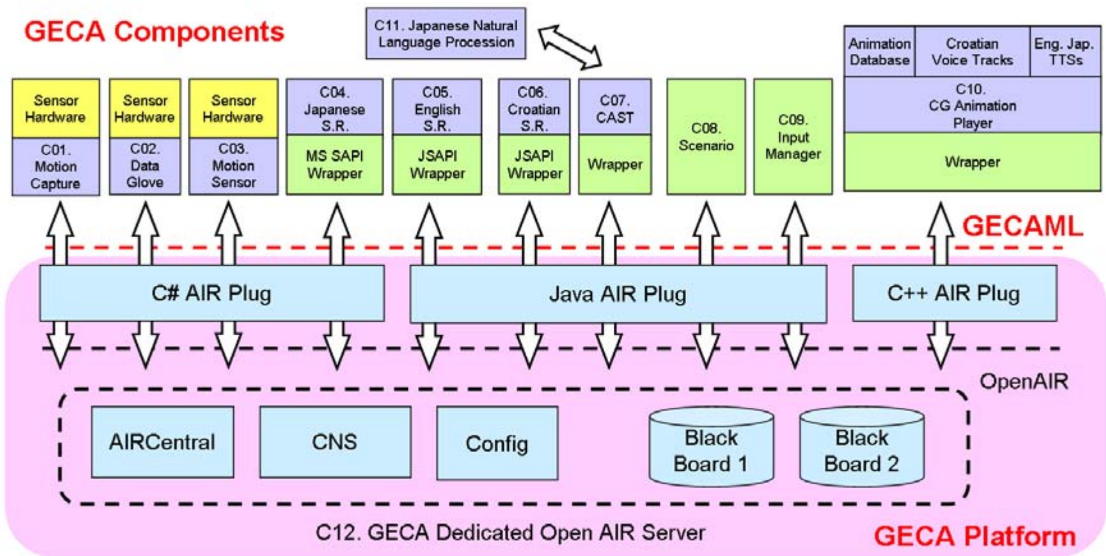

Figure 2. The conceptual diagram of the Generic ECA Framework and the configuration of the system

In order to achieve effective environment, the tour guide agent is projected on the large screen, standing as in real virtual space of city Dubrovnik. When a visitor comes to the system, the system recognizes the visitor is an English, Japanese or Croatian speaker from the speech recognizer's result. After the mode has been selected, agent continues to express his intentions in preferred cultural style, with both verbal and non-verbal channels. Example of scenario is: If user has been identified as Croatian, the agent then switches to its Croatian mode, that is, speaks Croatian and behaves like a Croatian to accept and answer the queries from the visitor while performing culture-dependent gestures according to predefined scenarios in that session. At the same time, the visitor can interact with the agent by both natural language speaking and also by non-verbal behaviors such as pointing to an object on the background image or raising his hand to indicate that he wants to ask a question.

During the eNTERFACE'06 period we could not manage to achieve all of the planned project objectives. At the begging of the workshop, since most of the team members were either Croatian or Japanese, we aimed to gather first-hand Japanese and Croatian cultural information where the differences are supposed to be fairly obvious. Unfortunately, because of the lack of good quality speech synthesizer and recognizer for Croatian, by the end of the workshop we have developed an agent that interacts with English instead of Croatian users. This was easy solution because most of non-verbal behaviors we implemented to demonstrate Croatian cultural style are also used in many western cultures speaking in English. Apart from that, we had general-purpose components to capture and generate English speech. At the end of the workshop, individual Croatian speech components were completed, but neither tested or integrated into the system. This was done in a further work.

Next two sections give an overview of a design and the concepts of our current system prototype by describing the architecture, concepts of the GECA framework, and various components we used to build an ECA agent system. Further, we describe possible extension of the system with agent that belongs to the different culture, and explain issues while making agent talk in Croatian and proposed solutions.

II. Concepts of the Generic Embodied Conversational Agent Framework

To connect many heterogeneous functional components of a ECA based system that can interact with human users in verbal and non-verbal way, there are several conditions indispensable: the consistency of all communication channels, timing synchronization of all components and handling streaming data from sensors in real-time.

GECA framework [20] contains three parts that serve those requires: it introduces a platform composed by a set of server programs for mediating and transporting data streams among ECA software components, specification of a high-level protocol based on XML messages and application programming interface for easy development of the wrappers for ECA software components.

Our platform is built upon a routing and communication protocol of cooperating A.I. programs, OpenAIR protocol [4]. The information exchange of ECA software components with XML messages mediates via and shared memory mechanism (blackboard or white boards in OpenAIR's context). It has the following advantages:

- Distributed computing model over network eases the integration of legacy systems
- Communication via XML messages eliminates the dependency on operating systems and programming languages
- Simple protocol using light weight messages reduces the computing and network traffic overhead
- Explicit timing management mechanism

• The use of shared backbone blackboards flatten the component hierarchy, shorten the decision making path and can realize reflexive behaviors

• Possible to use multiple logically isolated blackboards rather than traditional single blackboard

• Easy to switch or replace components which have the same function if they understand and generate messages in the same type

Figure 2 shows the conceptual diagram of the GECA framework and the configuration of the system. Based on this framework, we are specifying an XML based high-level protocol for the data exchanges between the components plugged into the GECA platform. Every GECA message belongs to a message type, for example, “input.speech.text”, “output.action.speak”, etc. Each message type has a specified set of XML elements and attributes, e.g., “intensity”, “duration”, “start_time”, etc. When a component starts, the message flow works like this; component registers its contact information (unique name, IP address, etc) to CNS (Central Naming Service) component and subscribes its interested message type(s) to the AIRCentral component. Then the messages in those types will be sent to the component from the specified blackboard (or a whiteboard in OpenAIR’s terminology) which behaves like a shared memory between the components when some other component published the messages. That component then processes the data it got and publishes its own output to the shared blackboard in certain message type.

By utilizing the communicative functions provided by the Air Plug libraries (currently we have developed the C#, C++ version libraries and a customized Java reference implementation from mindmakers.org) which are a part of the platform, an ECA system builder needs to develop a small piece program called a wrapper in order to handle and convert the input/output of an existing software component to be GSML (GECA Scenario Markup Language) compliant. After doing this, the heterogeneous nature of components that provide the same capability (both MS SAPI TTSs and JSAPI TTSs provide the same capability of the agent, i.e. to speak out from text) can be hided and behave identically to the other software components.

III. Tour Guide System components

Tour guide system is composed with the component configuration depicted in Figure 2. The following are the brief descriptions of those software components.

C01. Motion capture component. This component utilizes a simple motion capture device [5] using IR technology to roughly approximate a pre defined set of human visitor’s non-verbal behaviors.

C02. Data glove component. This component acquires data from a data glove hardware device and reports recognized movements of the visitor’s fingers to the other components.

C03. Motion sensor component. This component acquires data from a three dimensional acceleration sensor [6] which is attached on the visitor’s head to detect head shaking and nod movements.

C04. Japanese speech recognition component. This component is a wrapped MS SAPI5 [7] Japanese recognition engine which recognizes Japanese spoken by the visitors by matching predefined grammar rules and sends the recognized result as a text string to the subscribed components.

C05. English speech recognition component. This has the same function as component C04 but is a wrapped Java SAPI [8] speech recognition engine. Java SAPI is used here for keeping multiple speech recognizers on the same computer so that only one microphone is required to gather the user’s speech.

C06. Croatian speech recognition component. Because of the lack of a good enough speech recognizer for Croatian language, Croatian is recognized by an English speech recognizer with grammar rules which defined a limited range of Croatian vocabularies by approximating their pronunciations with English alphabets.

C07. Japanese spontaneous gesture generator. This component is a wrapper of CAST [9] engine that generates type and timing information of spontaneous gestures from Japanese text string by using a statistical model built from analyzing human presentation video.

C08. GSML scenario component. This component is an interpreter of GSML scenario scripts. It reads a script that specifies the agent’s verbal and nonverbal responses corresponding to specific behaviors from the visitors. This script is described in an XML based language representing Question and Answer pairs and is inspired from AIML [10]. A set of ECA specific capabilities distinguish it from AIML includes conversation state, nonverbal and verbal interactions, and objects in scene descriptions. All of the three language versions are described in the same script.

C09. Input manager component. The component combines sensor data streams which have no direct meanings to individual events representing human user’s nonverbal behaviors. Currently it combines data streams from motion capture component and data glove components and detects user’s nonverbal behaviors such as waving and pointing.

C10. CG Character animation player component. This component is a wrapped character animation player which is implemented with visage|SDK [11]. It accepts event messages from the other components and drives the CG character to speak or perform nonverbal behaviors. This player calls MS SAPI English and Japanese Text-To-Speech (TTS) engines installed in the same machine to generate the voice output of the agent and viseme events to drive the character animator to move the agent’s lips. Because of the lack of a good quality Croatian speech synthesizer, Croatian speech is done via prerecorded voice tracks and handled by the player. All of the nonverbal animations of the character are defined in an animation category included inside the player. The animation definitions in the category are not merely canned tracks but are parameterized and can be changed during runtime.

C11. Japanese natural language processing component. This component includes a Japanese morphology analyzer, Juman [12] and a Japanese syntax analyzer, KNP [13], the two analyzer label an input text string with linguistic information that will be used by C07. The communication between C11 and C07 is in their original way and is not included in GSML.

C12. GECA dedicated AIR server. This server mediates the communication among all of the components with a shared memory (blackboard) by subscribe-publication mechanism.

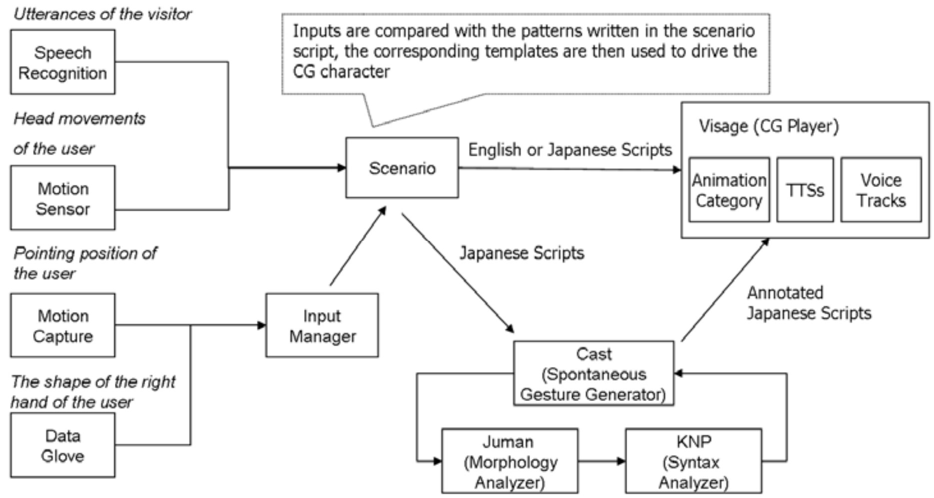

Figure 3. Data flow diagram of the ECA tour guide system

A key component for understanding and generation of the output actions of the agent (speech, body position, gestures) is GSML scenario component (C08). This component responds to a key words defined by speech recognition component (C04, C05), and non-verbal information from sensors (C01-C03) in the input of the system. According to the patterns written in scenario script it generates agent's behavior in the output.

Figure 3 shows the data flow among the components. Speech recognition component and sensor components (C01-C05) gather the verbal and nonverbal inputs from the human user. While speech recognition component gathers only specified key words, sensors collect all non-verbal inputs that are then recognized by input manager (C09). All results from input are processed to the scenario component (C08). The inputs from different modals are combined by the script interpreter and are matched with patterns – agent's behavior in the scenario script. It then sends the matched response which may include utterance and action tags to speech synthesizer and animation category component. The spontaneous gesture generator (C07) inserts action tags into the utterance according to the linguistic information from its natural language tagging companion. Scenario component sends messages written in a special XML format, GECAPL scenario XML format that defines agent's behavior. Phrases (speech text to Text-To-Speech component in character animator) and actions (visual agent action, such a nod or pointing) are wrapped into utterance tags and send via platform to the character animator (C10).

The character animator listens to action and speech events and plays them in real-time. Nonverbal animations are looked up from the animation category and filled with the parameters defined in the event messages. In order for the agent to behave differently for various styles (English, Japanese, Croatian), a large set of non-verbal animations is built. Current character animator (Visage) also provides TTS support. Moreover, it also provides a support for playing prerecorded Croatian voice files and matching animations. Furthermore, shortcuts between the sensor components and the animator that bypass the pipeline are allowed and make reflexive behaviors of the agent possible, and this is one of the strengths of this framework over the other ECA architectures.

A. *Culture dependent components*

When the system initializes, ECA agent is run in the English mode. Culture dependent information is provided after processing and recognizing verbal data captured from the input. If the system determines a presence of a visitor from another culture, it switches to Japanese or Croatian mode.

For language understanding and processing, two different components are used; speech recognition component for Japanese (C04) and English (C05). Croatian speech recognition component is the English component adopted for Croatian. The non-verbal behaviors are recognized by using the data from data gloves, infrared camera, and acceleration sensors (C01-C03). Nissho Electronics Super Glove is used to detect two statuses of a hand shape, pointing or five fingers straight. NaturalPoint OptiTrack FLEX3 infrared camera [6] detects spatial coordinates of a strap put around the wrist of a right hand. In this way, pointing, hand stability and waving are recognized. Nec/Tokin 3D Motion Sensor [6] tracks change of acceleration in three dimensions and we use it to detect two ways of a movement of user's head: nodding, a positive verbal answer, and shaking, the negative one. Above mentioned gestures, we are also implementing recognition of head orientation to approximate gaze direction and behavior to get the ECA's attention; raising one of the two hands.

Non-verbal behaviors detected within the system don't have importance in the multicultural domain. However, at the beginning of the eNTERFACE '06 workshop, we had intention to build a system that can smoothly distinguish user's cultural affiliation from his non-verbal cultural behaviors. Later we realized it is difficult to distinguish the cultural affiliation from only non-verbal inputs with the very limited hardware configuration which was not a problem of GECA framework itself. We found that there is no obvious general difference in non-verbal behaviors when people greet; a bow can be used to identify a Japanese user, but there is nearly no way to distinguish a greeting of an American and a Croatian user. In the end, we only use speech recognition as a key unit for determination of agent's cultural affiliation.

Output of the system is calculated by scenario component which uses GECA scenario script with patterns and matched responses for each language to generate agent's verbal and non-verbal behavior. Depending on the language definition, CG character player generates animation. If English or Japanese mode is selected, agent's speech is synthesized with TTS components, while for Croatian we use pre-recorded audio files and corresponding lip-sync animations. For each of three modes non-verbal animations are generated from animation database that contains cultural dependent gestures.

The most significant feature of a multicultural GECA based tour guide system is that provides an extension with another cultural domain. In example, we will explain the procedure if developer wants a tour guide agent to interact with French users.

Since non-verbal behaviors recognized in the system are not culture dependent, only one input component is required – a French speech recognition component. To deliberate agent's behavior, GSML based scenario script has to be extended with French significant patterns and templates to be processed in the system output. Further, to generate agent's speech and non-verbal behaviors, we could use TTS French engine and existing animations from animations category that fit French behaviors. Also, we could define new animations typical for French culture within the animations category.

Briefly, extension of the system with another culture demands following features;

- Adoption of a speech recognition engine, if present. It should be adopted to recognize key words defined by scenario script and should be wrapped into one of the AIRPlug libraries to be compatible with an OpenAIR's architecture. Otherwise, we propose a solution we use for Croatian version of the system.
- Existing GSML based scenario script should be extended with French agent's expressions that match the each input case.
- TTS engine. It has to be installed in machine where CG character player is run, and in player if French is processed, French-speaker should be selected.
- Existing animation category could be also extended with behaviors typical for French culture. Since most of the existing animations implemented in animation category are typical for western cultures (nodding for agreement, shaking head for disagreement) this issue is not essential.

The advantage of the wide-world speaking languages is assured presence of recognition/synthesizer engines. However, this isn't the case with Croatian. Since no recognizer/TTS engine for Croatian exists, we propose alternative solutions.

IV. Issues Concerning ECA Talking in Croatian and Proposed Solutions

Although Croatian is spoken by around 5 million people, commercial speech and language community has not yet produced general-purpose recognizers, synthesizers and translation engines for it. Therefore, a development of an ECA speaking Croatian requires a prior development of system components that can gather and recognize Croatian speech in input and generate speech of an agent as output. The following are existing solutions and solutions that were proposed during the eNTERFACE '06 period and were later integrated in the system.

A. *Croatian speech recognition*

Concerning the speech recognition for Croatian, some research has been done [18, 19], but neither of it had produced general-purpose recognizers. Therefore, it has been decided to modify an English speech recognition software component to recognize Croatian speech.

Following the scenario of the system, the pronunciation of Croatian words is approximated by using English alphabet. Since some Croatian words from scenario were impossible to write with an English alphabet and scenario was not that strict, we avoided them and used some other words instead.

Furthermore, we performed experiments with different Croatian speakers. For trained speaker's profile, word recognition results were satisfactory. However, if grammar contained similar words, those words were sometimes mixed by the recognizer, so it was better to choose words that are not so similar. For example, pronunciation of a Croatian word "da" (eng. "yes") is approximated with an English alphabet as "ddhaa". Although speech recognizer works well with recognition of word "da" in Croatian, it can sometimes mix that word with different words that contain word "da", like "slobodan" (eng. "free"). Therefore, it is better not to use short words e.g. "da" or "dan" (eng. day) that can appear in longer words.

In the end, because of the susceptibility in recognition of similar Croatian words and impossibility to approximate pronunciation of some words with an English alphabet, we had to make changes of the original English

TABLE I.
Croatian Words and Corresponding Grammar Rules

	Croatian word	Meaning in English	Approximation of Croatian pronunciation in English
1	Bok	Hello	bohk
2	Grad	City	ghraadh
3	Šetati	To go for a walk	shetthaatti
4	Fontana	Fountain	fonthaana
5	Pitka	Drinkable	peethka
6	Samostan	Monastery	saamostaan
7	Super	Super	supearh

scenario by defining new key words that have a different pronunciation in Croatian.

Table 1 shows Croatian words used for recognition and corresponding pronunciation of these words written by an English alphabet. Because only 5 scenes exist in our current system, transitions between the scenes and between the states in each scene do not require many key words from an input. In English and Japanese scenario we used 8 words for transition and in Croatian 7 of them. Later, while improving the system, we intend to increase the number of scenes and therefore it would be necessary to define new Croatian words for speech recognition by using grammar rules we created. However, because of limited number of the words that can be defined for recognition, we are also considering to research alternative solutions of this problem, e.g. instead of speech, to use classical mouse, keyboard or touch-screen as the system input.

B. *Croatian Speech Output*

In our system implementation, English and Japanese words that ECA speaks are generated by MS SAPI English and Japanese Text-To-Speech (TTS) engines. Concerning Croatian, only one product for speech synthesis exists; Wintalker Voice [15], developed by the community for education of blind persons UUOSSO and their Czech partners, Rosasoft [16]. Although this module works with both SAPI4 and SAPI5 and is very stable, it has a bad speaker quality. Our goal is to have comprehensible Croatian agent and since we have an automatic Lip Sync system [17] at our disposal, we chose to use a real person's recorded voice instead.

Our automatic lip sync system Lip Sync introduces a method for mapping a natural speech to the lip shape animation in the real time. The speech signal is derived from a type of cepstral representation of the audio clip, MFCC vectors, and classified into viseme classes using neural networks. The topology of neural networks is manually configured after series of experiments. When viseme is being calculated, it is send to CG Character animation player, Visage player [11] that uses MPEG-4 FBA International Standard for animation.

Once we had Croatian scenario, a native Croatian speaker has recorded speech the agent was supposed to say in certain situation in the noise free room. By applying a lip sync application we have created animations from the prepared speech files and made a database containing pairs of speech-animation files for every situation according to the scenario.

The database made during the eNTERFACE workshop is later integrated into the system. Integration included a modification of a script language GSML to describe action animations of non MS SAPI5 TTS engines and adoption of a CG Character player for playing those actions. Although solution with lip-sync animations is not flexible because database extension depends on the native Croatian speaker, the advantage is natural approach to human-computer interaction because of the quality of a human speech is far better than any synthetic voice and synthesizer.

V. Conclusion and Future Works

This paper presents a multicultural tour guide system based on GECA framework and issues while developing an ECA talking in Croatian; the drawback of Croatian recognition engine to determine user's verbal input and text-to-speech synthesis engine to generate agent's speech. We propose some solutions we made during the one-month period. However, our solutions are limited. We modified English speech recognition engine to recognize Croatian speech, but this component works well when key words that system recognizes don't match. On the other hand, to generate agent's speech we use database of prerecorded voices of a real human speech and lip-sync animations. This solution causes boundaries when expanding the system because database is speaker dependent.

Later improvements of Croatian version can include distinct research on speech/synthesis of Croatian. The optimal solution is to develop reliable speech recognition and good quality TTS engines. To build Croatian speech recognition module by we could consider using existing tools, like open source speech recognizer CMU Sphinx [14], or we could implement of our own system by using modern approaches such as HMM-based or ANN-based speech recognition methods. Development of a Croatian TTS engine can include adoption of existing engines for similar languages and involvement of research tools, like Open Text-To-Speech platform Epos [18]. However, the standing situation is difficult; the development of both general purpose systems demands large resources, several person-years, and is currently out of the scope of our research. For speech recognition, a small scale limited-vocabulary might be considered.

A multicultural system presents only beginning of the research on the cross-culture issue in the ECA-human interaction domain. We have started a joint project [21] with the University of Augsburg to explore the cross-culture issues in depth.

In general, a tour guide system is under development from a number of different directions. We are concerned on detection of culture dependent non-verbal behaviors. GSML language syntax is being improved to cover more complex situations in ECA-human conversations and we also intend to extend animation database with more expressions. Since tour guide system is used only for demonstrations, no evaluation has yet been done. However, demonstration we performed at the eNTERFACE '06 workshop woke up a high level of interest among human observers.

Acknowledgment

This work is part of a bilateral collaboration project "New Generation Embodied Collaboration Agents" between the Kyoto University and University of Zagreb. The work was partly carried out within the research project "Embodied Conversational Agents as interface for networked and mobile services" supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

[1] Cassell, J., Sullivan, J., Prevost, S., and Churchill, E.: Embodied Conversational Agents, MIT press, 2000.

[2] Huang, H., Masuda T., Cerekovic, A., Tarasenko K., Pandzic, I., Nakano, Y., and Nishida, T.: "Toward a Universal Platform for Integrating Embodied Conversational Agent Components", KES 2006 conference, p.p.220-226, Bournemouth, UK, October, 2006.

[3] www.enterface.net/enterface06

[4] OpenAIR protocol, http://www.mindmakers.org/openair/airPage.jsp

[5] NaturalPoint OptiTrack Flex 3, http://www.naturalpoint.com/optitrack/

[6] NEC/Tokin 3D motion sensor, http://www.nec-tokin.com/english/product/3d/index.html

[7] Microsoft Speech Application Programming Interface, http://www.microsoft.com/speech

[8] Java Speech Application Programming Interface, http://java.sun.com/products/java-media/speech/

[9] Nakano, Y., Okamoto, M., Kawahara, D., Li Q., Nishida, T.: Converting Text into Agent Animations: Assigning Gestures to Text, in "The Proceedings of The Human Language Technology Conference" (HLT-NAACL04), 2004.

[10] AIML (Artificial Intelligence Markup Language), http://www.alicebot.org/

[11] visage|SDK, Visage player, Visage Technologies, http://www.visagetechnologies.com/index.html

[12] Juman, http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

[13] KNP, http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

[14] http://cmusphinx.sourceforge.net/webpage/html/cmusphinx.php

[15] http://www.uuosso.hr/

[16] http://www.rosasoft.cz/

[17] Zoric, G., Pandzic, I.S.: A Real-time Language Independent Lip Synchronization Method Using a Genetic Algorithm, in "The Proceedings of ICME 2005, 6-8 July 2005".

[18] http://epos.ure.cas.cz/

[19] S. Ipšič, J. Žanert, I. Ipšić, F. Mihelič, "Speech Recognition of Croatian and Slovenian Weather Forecast", 4th EURASIP Conference, p.p. 637 – 642, July, 2003

[20] Peic, R. "A speech recognition algorithm based on the features of Croatian language", 4th EURASIP Conference, p.p. 613 – 618, July, 2003

[21] Culture-adaptive Behavior Generation for interactions with embodied conversational agents (CUBE-G), http://mm-werkstatt.informatik.uni-augsburg.de/projects/cube-g/