Doctoral Dissertation

A Generic Framework for Embodied Conversational Agent Development and its Applications

HUANG, Hung-Hsuan

Doctoral Dissertation

A Generic Framework for Embodied Conversational Agent Development and its Applications

Department of Intelligence Science and Technology

Graduate School of Informatics

Kyoto University

HUANG, Hung-Hsuan

Supervisor: Prof. NISHIDA, Toyoaki
Review Committee: Prof. KAWAHARA, Tatsuya
Prof. KUROHASHI, Sadao

September 2009

Preface

Embodied conversational agents (ECAs) are life-like computer graphics characters who can engage face-to-face conversations with human users and are ideal candidates of the interfaces for public services. This thesis describes three contributions to the development and applications of ECAs.

First, the Generic Embodied Conversational Agent (GECA) development framework is proposed. It integrates distributed and reusable ECA modules to behave as an integral agent. It is composed with three parts. GECA Platform is a network communication middleware based on a blackboard and XML message exchanging. It provides services including naming service, message subscription, and message forwarding management. GECA Plugs are the libraries that absorb the differences among operating systems and programming languages to facilitate the development of the wrappers of individual ECA components. GECA Protocol (GECAP) is a specification of XML message types and formats that are exchanged among the components. Based on this framework, GECA Scenario Markup Language (GSML) describing human-agent interactions and its execution component were developed to supplement GECAP. It is an XML-based script language to define a state transition model for a multi-modal dialog between the user and the agent.

Second, a couple of novel ECA systems for multi-user conversation have been developed by using the GECA framework. A multi-culture tour guide agent and a quizmaster agent are developed as the example GECA based systems. The quizmaster agent is deployed in real-world exhibitions and is further improved in the aspect of user attentiveness in multi-user situation which is typical in public exhibitions. Multi-user attentiveness is realized by two methods, one is rule-based and the other one is learning based with video / audio information acquired from the users' activities. Subject experiments are contacted and these two

implementations are evaluated by a quantitative psychology test, questionnaires, and user reaction analysis respectively. The results showed that measuring on the users' activities to decide the timing of the agents' actions can result in users' positive impressions during the interactions with the agents.

Third, a visual knowledge management system (VKMS), Gallery for large story-telling image / text collections is developed to assist content production for ECAs. The contents are represented as image thumbnails on a 2D zoomable surface and are logically organized by the user's direct manipulation. From the results of a subject evaluation experiment, it is shown to be effective in contents retrievals.

Acknowledgments

At first, I would like to thank my adviser, Prof. Nishida for his continuous and long-term supports especially on research directions and the techniques of paper writing. Also thank to Prof. Kawahara and Prof. Kurohashi who gave me many valuable comments in the preliminary defense, so that I could further improve the dissertation.

Many thanks to Prof. Pandžić and Prof. Nakano for the very fruitful discussions and advices in the monthly video meetings and the two eNTERFACE workshops. Also thank to them for the software tools from their laboratories.

A lot of thanks to all of my colleagues in Nishida-Sumi Lab. of Kyoto University. Without the great contributions from my project members, Furukawa-kun, Ohashi-kun, Iwaki-kun, Inoue-kun, Masuda-kun, and Katya, the works included in this dissertation were not possible. Dr. Kubota always gives me precise and helpful advices. Mrs. Komatani and Miss Tsukada always help me kindly and listen to my thoughtless requests. Prof. Sumi gives me many useful comments in the meetings and presentations.

Thanks to my trustworthy partners in the University of Zagreb. Aleksandra is my first-place partner for four years. Vjekoslav and Goranka helped a lot in the eNTERFACE'06 workshop. Thanks to Yamaoka-kun of TUAT for his great contribution in the eNTERFACE'08 workshop. Thanks to Dr. Magariyama, Mrs. Shichiri, Kanai-san and many other staff of NFRI for their best support in the quiz agent exhibitions.

Thanks to my family who supported and tolerated me for so many years staying abroad.

Finally, I would like to particularly thank to Kondo-sensei of Yamasa Institute. Without her kind and generous help, I could never get into Kyoto University and could not become a researcher.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machines can work without rest while keep constant and highly accurate quality which can never be achieved by humans. For decades, artificial intelligence researchers pursue the ultimate goal to build the machines which can engage the conversation with humans at a level close to human-human one. The concept video, *Knowledge Navigator* produced by Apple in 1987 was a good example of this idea. The term, Embodied Conversational Agents (ECAs) is first proposed and defined by (Cassell, Sullivan, et al., 2000) as "*computer interfaces that can hold up their end of the conversation, interfaces that realize conversational behaviors as a function of the demands of dialogue and also as a function of emotion, personality, and social conversation.*" ECAs are usually realized as life-like characters in 3D computer graphics animation (hereafter ECAs) and are the center of this dissertation.

In face-to-face conversations, we humans not only use language but also fully utilize our body to communicate with the interlocutors. We adjust the tone of our voice according to the context of conversation, perform hand gestures (Kendon, 2004), change body postures to supple speech, and monitor those expressed by the interlocutors at the same time. In order to achieve these conversational functions on a machine, sensors are required to perceive verbal and nonverbal status of the human communication partners, and actuators are required to realize the agents' intentions as perceivable behaviors to humans. The difficulties do not only come from what the agent can do but also come from the subtle differences of the quality of their movements and their appearance. Comparing to ECAs' mechanical counterpart, humanoid communication robots, they have the potential advantages in larger

degrees of freedom in their faces and bodies, less noises in actuation, and less limitations in the virtual environment where they are.

Despite the lack of physical actuators and the same difficulty in perception processing, ECAs relieve researchers from mechanical and material issues with the relatively lower hurdle in rendering and animating computer graphic characters realistically. This allow them to concentrate on realizing high-level and advanced conversational abilities like speech synchronized lip movements, rich facial expressions with synchronized and sophistical movements involving all parts of the face. Therefore, ECAs can be considered as ideal interfaces for applications such as the simulations in psychology studies, language training, entertainment purposes, or public services where high-level communication abilities are required.

1.1 Contemporary ECAs

With the advance of computer hardware, computer graphics, natural language processing, speech recognition and synthesis technologies, ECA attracts great interests from researchers in the past decade (Prendinger & Ishizuka, 2004; Nishida, 2007), and ECA systems in a diversity have been developed. For example, Rea (Real Estate Agent) (Cassell et al., 1999; Cassell, Bickmore, et al., 2000) is an ECA who mediates house information with single user. Rea uses simple heuristics on verbal and nonverbal behaviors done by the user to do conversational turn management, she yields the turn to the user when the user starts speaking and terminate her own utterance in the middle when the user starts to do gestures. Herself also does synchronized multi-modal utterances. MACK (Media lab Autonomous Conversational Kiosk) (Cassell et al., 2002; Y. I. Nakano et al., 2003) is an ECA who can answer questions about and give directions to the MIT Media Lab's research groups, projects and people. MACK uses a combination of speech, gesture, and the indications on a map placed on a table between himself and individual users. The users' head movements and gaze directions are tracked by MACK for him to estimate whether the user has understood what he just said (grounded) and to decide whether to proceed or explained in more detail. Greta (Pelachaud et al., 2002) is a doctor agent who gives her patients information about a drug prescriptions. She is implemented as a 3D talking head and has her own personality and a social role, and the capability of expressing emotions, consistently with the conversation context with her

own goals. Max (Multimodal Assembly eXpert) is a virtual human developed in Bielefeld University and is adopted as various roles with different abilities. As a assistant of human user to collaboratively construct virtual objects (Kopp & Jung, 2000; Kopp et al., 2003) with multi-modal interaction, a master of a card game with emotion simulation (Becker, Nakasone, et al., 2005; Becker, Prendinger, et al., 2005; Boukricha et al., 2007), and a science museum guide (Kopp et al., 2005; Kopp, Allwood, et al., 2008) with real-time feedbacks to visitors' keyboard inputs.

1.2 The Need of a General Purpose Framework

In order to realize a believable ECA capable to take out natural face-to-face and multimodal conversation with humans is not easy. In addition to the prosody properties of verbal channel, precise control on non-verbal channels like gazing, raising of eyebrows, nod, hand gestures or postures in performing communicational functions like directing the flow of conversations or as an supplement of verbal utterances while appropriately reflecting the agent's internal emotional state, personality and social status as the response to recognized attention of human users with sensing devices. Finally, output with realistically rendered characters, environment as well as fluent speech synthesis. To realize these abilities with a software agent, the knowledge and techniques on signal processing, natural language processing, gesture recognition, artificial intelligence, dialog management, personality and emotion modeling, natural language generation, gesture generation, CG character animation and so on are required.

ECA involves many research disciplines so that it is difficult for individual research teams to develop from scratch. No matter what field a developer who is going to build an ECA is in, (s)he needs to include a minimum set of these functionalities into his (her) ECA. The usual way to build ECA systems is therefore by utilizing software tools developed by other research groups. However, because of software tools developed by different institutes are neither meant to cooperate with each other nor designed for the same application domain, usually it is laborious or even impossible to make them work with each other. More than that, redundant efforts and similar approaches are repeated by the researchers due to their common needs.

To solve these problems, if there was a common framework that absorbs the heterogeneities to connect diverse ECA software tools and drives the connected components as an integral ECA system, redundant efforts and resource uses can be saved. Furthermore, the sharing of research results can be facilitated and the development of ECA systems can become easier.

1.3 Culture-enabled Interface as an ECA Application

The recent advances in transport and communication technologies have globalized markets and businesses and have changed the way people interact with each other. Enterprises pursue success in overseas markets to maintain their competitiveness, and businessmen have to negotiate with their foreign customers. In the academic world, attending international conferences is the most efficient way for researchers to gather first-hand information. Overseas trips for tourism and other personal reasons are also becoming easier and more popular. The ability to communicate face-to-face with people who come from other cultural backgrounds is gaining importance.

In order to consider the cultural issues in computer-human interfaces, depending on the needs of the application, there are two approaches: internationalization and localization (Young, 2008). Internationalized designs exclude culture-dependent features and implement behavior that will be interpreted in the same by people from different cultures and prevent misunderstanding. Localization includes culture-specific designs for the target audience. According to research reports such as that of (Nass et al., 2000), people prefer interface agents with the same ethnicities as themselves; they feel more comfortable with and tend to be more trusting of these agents. (Baylor et al., 2006) investigated the impact of the appearance of an interface agent in terms of the age, gender, and "coolness," and reported that participants prefer peer-like (similar to the participants) agents. (Pickering & Garrod, 2004, 2006) reported that people tend to align their use of language to the interlocutor during dialogues. This alignment is the basis of successful communication. (Costa et al., 2008) suggested that speaking in a second language could impair the alignment in dialogues. In the case of an interface agent for users who may come from many cultural areas, such as a tour guide agent for a sightseeing spot, information transfer should be more efficient if the

agent speaks the user's native language and shows behaviors familiar to the user.

Inter-culturally competent ECA system development typically applies the classic "analysis by synthesis" method:

1. Conduct data acquisition experiments and observe human-to-human interactions.

2. Hypothesize the principal requirements for human-agent interactions and implement a prototype system.

3. Analyze the prototype system and verify the hypotheses; if the results are not satisfying, then go back to step 2.

In this development style, the researchers can clearly benefit if the system can be partially replaced and prototyped rapidly.

1.4 The Deployment of ECAs as Real-world Applications

From ECAs' inherent characteristic, they are ideal candidates of the interfaces for public service systems because the users can use the communicational skills what they are used to in their daily life without prerequisite training. However, due to the issues resulted from the installation of ECA systems in public spaces like:

- Limited sensor device and technology usages come from the more noisy and unpredictable environment.

- Higher requirements on robustness and intuitiveness of the interface because of the untrained users.

Most ECA research works were conducted in laboratories where the devices can be specialized, the environments can be fully controlled, and the users can be instructed or trained.

1.5 The Content Management Issue

Current trend of ECA research is task/functionality oriented, i.e. researchers are pursuing the improvement of ECAs' abilities rather than the contents of the ECA system. Nowadays, the content of ECA systems or the internal knowledge of the ECAs rely on hand-coded rules defined by the developers. ECAs can usually be divided into two categories according to the rules, chatbot (Weizenbaum, 1966) style , i.e. reactive to users' utterance without intention, or an agent planning the actions which can achieve its goals with a BDI (Belief-Desire-Intention) engine. The size of the rule set varies from several hundreds (e.g. the Max agent) to several dozens of thousands (e.g. ALICE bot) (A.L.I.C.E. AI Fnd., 2005) of rules. In order to build attractive systems, rich content is required. When the size of the content collection becomes large, its management becomes a critical issue.

Pictures and text are counterparts that are the most basic yet essential media for knowledge dissemination. In some cases, a picture tells a story worth more than a thousand words, and in other cases, even a single word cannot be represented by any picture. The invention of the camera in 1839 drastically changed the manner in which people recorded their memories. A photograph is not merely a snapshot; it also tells a story in its background. Nowadays, photographs have become an indispensable medium for knowledge dissemination. In recent years, digital image acquisition equipments such as digital cameras, scanners, and video cameras have evolved and their prices have dropped very rapidly. In particular, digital cameras have become ubiquitous and are fast replacing traditional film cameras. The advantages of the digital camera are its near zero running cost and the immediate preview of the image obtained. Therefore, people are beginning to show greater willingness to record their daily memories in digital photographs. Even a non-keen photographer can now easily accumulate thousands of photos within a short time using a digital camera. For example, one of the authors has captured more than five thousand photos per year using a digital camera. Further, the widespread use of the Web makes it very easy to obtain image information. Consequently, personal digital image collections have grown very rapidly, and the problem of managing large collections has emerged, which is growing in importance with each passing day.

1.6 The Contributions of this Dissertation

This dissertation includes the following research contributions that address the issues mentioned in previous sections:

- The proposal of a general purpose framework for ECA development what is not available yet in ECA research field. It is called Generic ECA (GECA) framework and includes an integration platform, a set of API libraries, and a reference starter toolkit of essential components of a fully operational ECA. The reference implementation can then be modified or extended for different purposes.

- A multi-culture adaptive agent named Dubravka is developed as an example application and the testbed of GECA framework. Multi-modal human-agent interaction and multi-user setting are investigated, the experiences can be utilized for the development of more advanced ECA systems.

- A simple quiz game agent which has no user-awareness has been developed and deployed in actual exhibitions. This agent is latter improved to include user attentiveness in multi-participant situation where is typical in public exhibitions but is not thoroughly investigated yet in ECA research field. The improved agents are then thoroughly evaluated with the combination of objective psychological method, questionnaires, and video analysis rather than the other works that are usually only evaluated with subjective questionnaires.

- The concept of knowledge management systems with visualized content is studied. We call them Visual Knowledge Management System (VKMS) and propose one implementation, Gallery system. In Gallery, a piece of knowledge is represented as a picture complement with a text segment what we call a *knowledge card* (Kubota et al., 2004). Knowledge cards then can be linked as *stories* that can be presented by ECAs.

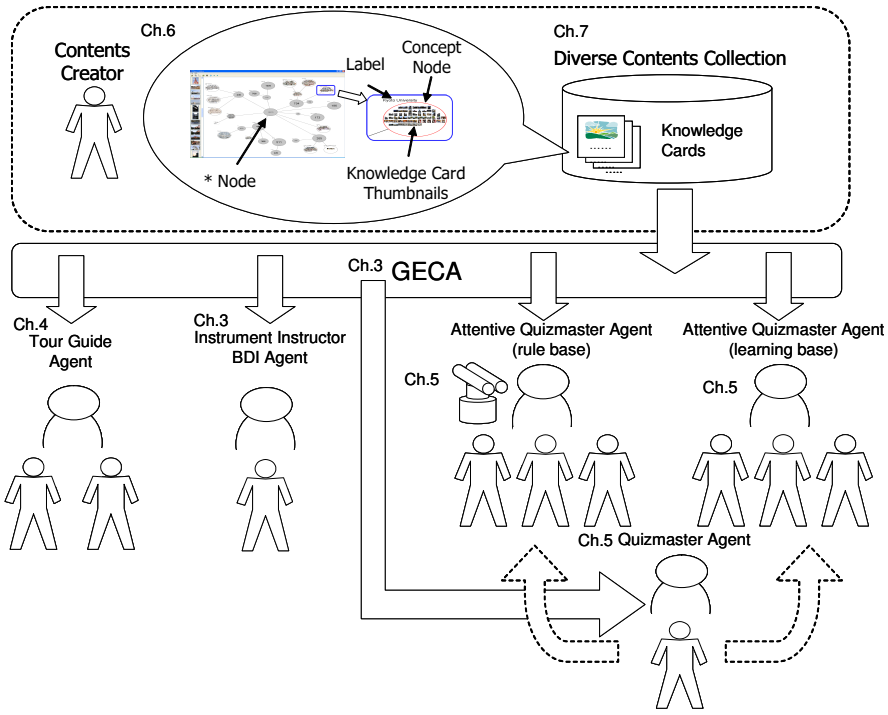

Figure 1.1: The main contributions of this dissertation and their relationship between each other

1.7 The Organization of this Dissertation

Figure 1.1 depicts the relationship between the contributions of this dissertation. This dissertation is organized as following chapters: Chapter 2 reviews the work related to the topics of this dissertation. Chapter 3 introduces Generic Embodied Conversational Agent (GECA) framework that is a programming framework to ease the development of embodied conversational agent systems. Chapter 4 introduces a virtual tour guide agent developed as an example of GECA applications. Chapter 5 introduces the real-world deployable quiz agents in detail. Chapter 6 introduces Gallery, a system that is designed to efficiently store, manage and reuse large amount of knowledge contents. Chapter 7 discusses the critiques and future works of this dissertation. Chapter 8 concludes this dissertation. The follows are the short summaries of main chapters of this dissertation.

Chapter 2

This chapter reviews the works related the topics of this dissertation. Because the emerging research interests on virtual human animations and the demands for standardization of ECAs, there are already a number of virtual character description languages proposed by several individual research institutes. But none of them got to be widely accepted and could become a common standard. A cross-institute joint research team has started the development of BML (Behavior Markup Language). A BML specification draft and a prototype of BML inverse kinematics converter are available, but many parts of it are still unclear or missing. There are several ECAs have been deployed in exhibitions or museums, but the main research concern of most of these works are on user reaction analysis in simple human-agent interactions, few of them included the ability to be attentive to the dynamically changing user activities in multi-user situations which are typical in public exhibitions. The knowledge contents of ECA system are usually relied on hand coded rules and are not rich. The content management system dedicated for large size internal knowledge of ECAs is not yet available. On the other hand, the techniques in information visualization, idea generation support and image repository management provide the hints of effectiveness of spatial memory and zoomable interface.

Chapter 3

This chapter discusses the issues emerged in a general purpose framework for developing ECAs. It then proposes the basis of this dissertation, Generic ECA framework. This framework is composed of a low-level communication platform, a set of communication API libraries, and a high-level protocol. The integration platform is a network communication middleware based on a blackboard and XML message exchanging. It provides services including naming service, message subscription, and message forwarding management. The libraries absorb the differences among operating systems and programming languages to facilitate the development of the wrappers of individual ECA components. The protocol is a specification of XML message types and formats that are exchanged among the components running on the platform. GECA Scenario Markup Language (GSML) describing human-agent interactions and its execution component were developed to supplement

GECA. GSML is an XML-based script language to define a state transition model for a multi-modal dialog between the user and the agent. The development of the first GECA server prototype as well as .Net, C++, and Java versions of libraries have been completed. We have also implemented several essential components for general purpose use.

Chapter 4

This chapter presents the development of two example GECA based applications to show the usefulness of GECA and to explore the general issues in developing ECAs with GECA. The goal of the first project is to develop a virtual tour guide called Dubravka who mediates sightseeing information and serves its users either in Japanese, Croatian, or general Western cultural modes. Users can use multiple modalities including speech, pointing gesture, and head movements to interact with the agent. The cultural modes distinguish to each other in speech input/output and the nonverbal behaviors of the agent. This system is basically implemented as a four-week student project in the eNTERFACE'06 workshop. The development could be done fast because the benefits from the modular design of GECA, interchangeable and reusable components. This tour guide agent is further extended to be able to interact with two users in the eNTERFACE'08 workshop. Multi-user agent interaction issues including dynamically changing user number, the conversations between the users and attention recovering are investigated.

Chapter 5

This chapter introduces a series of studies on the improvements of the attentiveness toward multiple participants of the NFRI quizmaster agent. The first simple prototype without user awareness is deployed in four exhibitions since 2007. From the exhibitions of it, we found that the visitors usually come to the exhibitions in groups, they usually discuss with each other to solve the quizzes, and the activeness of their discussions changes dynamically in the game sessions. Two approaches are then proposed to improve the agent's life-likeness by incorporating user attentiveness functionalities in multi-user situations. One aims to achieve an utterance policy by determining appropriate utterance timing and addressee from the participants' status. The measurement is done by tracking the participants' face movements

and the activeness of their conversation from audio information. The other approach introduces a transition state model of the agent's attitude toward the participants' status. The state transitions are learned with a support vector machine (SVM) classifier by using video and audio information from a video corpus collected in a Wizard-of-Oz (WOZ) experiment. This transition model drives the agent's idle motions and utterances in expressing its attitude varying from anxious to impatient toward the participants' status. To evaluate these two prototype systems, we used of a method called GNAT (Go/No-go Task) test. It is an objective measurement of the participants' implicit attitude (e.g. natural) toward certain attribute on certain concept (e.g. an agent). The evaluation process is complemented with regular questionnaires and video data analysis. By combining these results, we concluded that these two approaches do have positive influences on the participants' perceptions of the agents.

Chapter 6

This chapter describes a VKMS, Gallery. It features a zoomable 2D graphical space that represents a large storage of knowledge cards as a tree structure. Users can browse their repository there smoothly from overall view to individual cards. In this space, each card is shown as one or multiple image thumbnails that are contained in the concept nodes where the card's content coincides. Each concept node represents one thought of the user. Users build their own knowledge space by generating descendant nodes from the root node where all cards are in with filtering conditions like annotated keywords, file paths, and modified date. The conditions can be logically combined with each other. To utilize humans' spatial memory what is considered as effective in management, these operations and the special layout are done by the users with direct manipulations like drag-and-drop. Two subject experiments are conducted to evaluate the Gallery system. One is a two-week usage analysis in the aspects of the characteristics of the knowledge space built by different participants. The other experiment is the comparison on the efficiency of information retrieve after a two-week period with a well-known commercial image manager. From the experiment results, Gallery is proved to be more effective in memorizing what the images are in a collection.

Chapter 7

This chapter discusses the critiques of the works described in previous chapters and possible future works to improve them. These include the support of BML in GECA, the realization of multi-party conversation in the Dubravka agent, the integration of the two approaches proposed in chapter 5, Web based GECA agents, and the integration of Gallery with GECA. This chapter ends with a proposal of the Circulating Knowledge with Virtual Agents (CINOVA) framework that aims to facilitate the knowledge circulation process between institutions and their public audiences. It is composed with three main parts, VKMSs, embodied conversational agents for interactive knowledge presentations, and a shared knowledge repository. The data flow of this framework is as the follows: the experts in the institute provide knowledge to the shared repository, the creators reorganize the knowledge to create presentations contents, virtual agents present those contents to the visitors in exhibitions or on the Web, and the users feedbacks their information queries to the creators through the interactions with the agents. The unit of common knowledge presentation is a knowledge card that is composed with an image and descriptive text. The presentation contents are composed as sequences of knowledge cards, or stories. Required extended works include the realization of the shared knowledge repository, rich common knowledge representation, and interactive web-based GECA agents.

Chapter 2

Related Works

This chapter reviews current status of the works related to the topics addressed in this dissertation. The activity of the standardization of ECAs is described in section 2.1. The ECA applications which are related to the ones studied in this dissertation is introduced in section 2.2. The works related to content management or photo management are introduced in section 2.3.

2.1 Standardization of ECA Development

Because the emerging research interests on virtual human animations and the demands for standardization of ECAs, there are already a number of activities trying to standardize the production of CG characters or autonomous ECAs. In this section, we introduce them in two categories. First, the activities in attempting to propose a standard description language of character animations. Second, a being developed framework that is meant to address the standardization of the behaviors expressed by autonomous ECAs.

2.1.1 Character Animation Description Languages

Some high-level conversational agent or virtual human description markup languages have been proposed or are being developed such as AML (Avatar Markup Language) (Kshirsagar et al., 2002), VHML (Virtual Human Markup Language) (Gustavsson et al., 2001), CML

(Character Markup Language) (Arafa & Mamdani, 2003), APML (Affective Presentation Markup Language) (Carolis et al., 2001), and MURML (Multimodal Utterance Representation Markup Language) (Kranstedt et al., 2002). AML is a high-level script language specifying avatar animations; the AML processor reads AML scripts containing high-level descriptions of avatar facial expressions, body animation, and utterance text of the avatar, or references to MPEG-4 FBAP (Facial Body Animation Parameter) (ISO/IEC JTC1, 1999; Pandzic & Forchheimer, 2002) files, and then it generates the corresponding MPEG-4 bit stream for Web based applications. However, the agent architecture is deterministic and thus has no flexibility; the script language does not consider the input part from the human user, either. VHML is a high-level markup language that describes a virtual human for general purposes, it is composed with a set of sublanguage includes descriptions on emotion, facial expressions, and gestures, etc. However, the specification of VHML is distinct and thus has little flexibility to include supplement FAP/BAP files like AML does. Many parts of it are still undefined, especially the gesture or body animation parts. CML is another under-development high-level virtual character description markup language which is similar to AML. It differs to AML with the specification of emotion and personality model while its predefined base set of movements can not be extended dynamically. APML is a language that specifies the association of verbal utterance, facial expression, and dialog moves (Traum et al., 1999) of a talking head agent. MURML associates gestures with begin/end timing marks that are inserted into verbal utterances. Each gesture is described with a set of parameters presenting wrist location, hand shape, and wrist orientation.

MPML (Multimodal Presentation Markup Language) (Prendinger et al., 2002), MPML-VR (Okazaki et al., 2002) and TVML (TV program Making Language) (NHK , 2009) are very high-level script languages designed for easy making of presentation or TV-program like contents. With their user friendly interfaces, these contents can be created by writing a simple script to describe a limited predefined set of virtual word, objects, characters, and character behaviors.

The activity is intensive, but none of them got to be widely accepted and became a de facto standard. This can be considered to be due to the following reasons:

It is difficult to find a balanced abstractness and thorough coverage of a high-level description language. There are virtually infinite possible behavior can be done by human

and so as the human-like characters. Character animations which are considered as look natural vary from application to application, character to character. Therefore, in most of cases, the languages can only to be specified as extremely high level where concrete specification could be figured out. This limits the benefits to adopt such a language rather than a home-made description language which is most suitable to the researchers' own purpose. The same reason also resulted in the fact that most of these languages are similar to each other but no one of them dominates the remainings.

The lack of compliant character animation toolkit bundled. Most of the aforementioned works do not provide a fully functioning character animation toolkit except MPML dialects and TVML. If a description language neither specifies the animations concretely nor provides a animation toolkit, it is hardly to be useful for ECA developers. On the other hand, although MPML and TVML provide easy-to-use and fully functioning toolkits, they can not be extended easily and thus their application is limited.

In contrary to the languages mentioned above, MPEG-4 FBAP is a specification trying to achieve video communication of conversation partners with avatar animations through a narrow network channel. Detailed character animation parameters are specified in this standard, where the CG character is animated in a way like a virtual robot, i.e. rotating the joints in the sense of angles. A VRML97 (Virtual Reality Modeling Language) (Web3D Consortium, 1997) based representation standard of humanoid model, H-Anim (H-Anim WG, 2002) is adopted. There are 66 low-level and two high-level (expressions and visimes) parameters specified for the facial animations as well as 296 parameters specified for the body animation. In this way, the ECA developers have to calculate inverse kinematics to animate the character. Some software packages are available for MPEG-4 FBAP, for example, the visage|SDK (Visage Technologies AB, 2008) used in this study.

2.1.2 SAIBA Framework

To scaffold ECA production process and encourage sharing and collaboration, a group of ECA researchers has initiated a work called SAIBA framework (Situation, Agent, Intention, Behavior, Animation, (mindmakers.org, 2006)). The framework specifies multimodal generation and consists of processing stages in three different levels (Figure 2.1):

Figure 2.1: The conceptual diagram of SAIBA framework (from (mindmakers.org, 2006))

1. planning of a communicative intent
2. planning of a multimodal realization of this intent
3. realization of the planned behaviors

This working group aims to provide two common languages for describing ECAs. One serves as the interface between stage 1 and 2 what they call Function Markup Language (FML). The other one is the interface between stage 2 and 3 what they call Behavior Markup Language (BML).

FML

FML is a language that describes communicative and expressive intention of ECA without any reference to physical behavior. It is meant to provide a semantic description that accounts for the aspects that are relevant and influential in the planning of verbal and non-verbal behaviors. The specification of FML is still in its very beginning stage. The first FML workshop is held together with the AAMAS 2008 conference where the author of this dissertation also attended. In this workshop, the researchers discussed the range that FML should cover: what does FML actually mean? what does the term *intention* mean? should *culture*, *emotion*, *personality*, or *context* be included as well? The discussion was started from a very abstract view and there was no concrete agreement achieved in this workshop, but the researchers concluded to form smaller groups to develop proposals based on four specific scenarios. These scenarios include:

- Dyadic conversation with a human in a scenario where the agent is collaborating with the user on the construction of a physical object. The negotiations involving the topics like what to do in next step to achieve the goal are expected.

- Presentation agent presenting a science exhibit to visitors at a science museum. It is considered to be a "long" monologue, and the agent is assumed not be able to perceive the audience feedback.

- Multi-party conversation in social interactions expected to happen in a restaurant. The participants of the conversation is assumed to be dynamic, i.e. the participants may join and leave the conversation during it is being taken.

- Long term companion agent in the health domain. The scenario will describes two to three interactions at widely separated points in time during this long-term relationship.

BML

BML is a language meant to describe multimodal behaviors as they are to be realized by the final stage of the generation process. It provides a general, player-independent description of multimodal behavior that can be used to control an ECA. In contrary to FML, the aspects where BML is aimed to address are much more concrete. The working group first proposed the idea in (Kopp et al., 2006) and discussed their progress and some specific technical issues in (Vilhjálmsson et al., 2007). A draft specification (Mindmakers.org, 2008) has been published.

It distinguishes to the other languages that are introduced in section 2.1 in mainly proposing the syntax describing the synchronization of the multiple modalities of the character. In BML, a concept called "synch point" is proposed. Each individual nonverbal action of the character has a single *ID* and six phases which are divided by five points, *Start*, *Ready*, *Stroke-start*, *Stroke-end*, *Relax*, and *End*. Speech texts are inserted with synchronization marks. The synchronization of multi-modal animation is then described via the alignment of these synch points by referencing the action IDs. In BML, the working group defined the character animations as the following core categories: *posture*, *locomotion*, *speech*, *gesture*, *face*, *head*, and *gaze*. Each category has its own set of XML elements and attributes and has a minimum set of animations which must be implemented by any BML compliant player that the developers call the *level 0* of BML. The BML activity is still in its progress and the specification is changing. Many parts are still missing or left unclear, e.g. locomotions which has a target like walking and facial expressions. Currently, facial expressions seem

going to be specified with detailed parameters based on FACS (Facial Action Coding System) (Ekman et al., 2002). This is an incoherent style to nonverbal animation specifications in BML what are mere specified with abstract names like `nodding` and `shaking` of head. Although the BML specification is not completed yet, several institutes have started the works related to it. The ECA team in South California University has developed a BML compliant inverse-kinematics engine called SmartBody (Thiebaux et al., 2008). The team in Reykjavík University developed a BML realizer by combining SmartBody and the free 3D graphics engine Panda3D (CMU, 2009) developed by Carnegie Mellon University.

2.2 ECA Applications

Because the emerging concern on ECAs, there are large numbers of ECA applications available. This section only includes a brief review on the ones which are related to the works of this dissertation. ECAs who engage cultural issues, ECAs deployed in real-world application, and the ones engage more than one users are introduced.

2.2.1 Cultural ECAs

The differences among cultures appear not only in languages and their use, but also in the display of internal emotional state in facial expressions, gestures, the range of movements, interpersonal distance, and so on (Isbister, 2004). Computer graphic characters or embodied conversational agents (ECAs) who can speak in the natural language and display rich facial expressions and who have large degrees of freedom in body movements are ideal interfaces for culture-enabled systems.

A number of research groups have studied the use of ECAs in immersive training and pedagogical applications for inter-cultural communication. Examples include the TLTS (Tactical Language Training System) project developed for training US soldiers in foreign languages and culture to smoothen the execution of their missions abroad (Johnson et al., 2005), an attempt to use virtual peers to encourage African American children to switch their language coding to increase school-based literacy (Iacobelli & Cassell, 2007), a proposal for modeling cultural differences as computational parameters based on a combination of the

analysis of a video corpus collected in experiments, and a theoretical model (Rehm, Nakano, et al., 2008).

2.2.2 Real-world Deployments of ECAs

Making ECAs go public in exhibitions or museums is an emerging challenge in recent years. Many of these research efforts focus on the analysis and classification of how the visitors interact with the agents in museums on a relatively long-term corpus, say from several months to years. For example, the Swedish free-talking virtual characters, August (Gustafson & Bell, 2000) and Pixie (Bell & Gustafson, 2003) who are installed in culture and telecommunication museums respectively. The authors further investigated how adults and children are different in trying to resolve system's errors in ASR (automatic speech recognition). Max (Kopp et al., 2005; Kopp, Allwood, et al., 2008) is a guide agent installation in a computer museum. He can perform real-time feedback behaviors from the visitors' keyboard inputs and responds to multiple visitors by image processing techniques. Sgt. Blackwell (Robinson et al., 2008) is a virtual human exhibited in a design museum. He answers free questions from the visitors without predefined conversational goals. Some of them are exhibited in computer expositions. For example, CrossTalk (Klesen et al., 2003) is an interactive theater with virtual actors exhibited in CeBIT 2002, and IEAS4Games (Gebhard et al., 2008) is a poker game that features two virtual characters with artificial emotion, mood and personalities and is exhibited in CeBIT 2008.

2.2.3 Multi-party Human-agent Interaction

Traum (Traum, 2003) provided a principal literature on general issues to realize multi-party human-agent interactions. They can be summarized as the follows.

Participants' role management. Unlike dyadic dialogs where there are only speaker and addressee, in multi-party dialogs, the identification of conversation participants' roles including addressee, overhearer and speaker is necessary.

Interaction management. Managing the communication flow in a multi-party dialog is more complex than dyadic dialogs because there are potentially more interlocutors to acquire dialog turns from and transfer dialog turns to. The management of the use of multiple

channels like speech and gestures also becomes more complex.

Topic, grounding and obligation management. In multi-party communication, there are more participants who may propose new topics and potentially more simultaneously opened topics. Therefore, the management of who is talking on what topic and should speak what to whom as well as what is the grounded truth with each interlocutor becomes more complex.

Most of contemporary ECA research works that address multi-party interaction issues focus on multi-agent / single-user configurations. For example, a car presentation team consisting of a salesman agent and a customer agent (Andre & Rist, 2001), a tactical training system for the soldiers who are going to be deployed abroad (Traum et al., 2003), cellular phone presentations via the discourses between two agents who are attentive to the gaze direction of the user (Eichner et al., 2007) and so on. FRED (Vertegaal et al., 2001) systems is another example studying on how humans shift gaze directions among two virtual characters depending on conversational status, the authors then proposed a statistical model of gazing directions during multi-party conversations.

On the other hand, in multi-user configurations, the conversation situation is more unpredictable and thus more difficult to be realized Gamble (Rehm et al., 2005) is a dice game where an agent interacts with two human players. The round based game rules fixed the system's scenario and resulted in basically three dyadic interactions. By using the same system, Rehm and Andre (Rehm & Andre, 2005; Rehm, 2008) found the human players' mixed behaviors interacting with the agent or the other player in the game. The human players showed similar reactions to the agent as what they do to the other player but also some behaviors what are considered as impolite or rude, for example, they showed the gazing patterns to spend more time in looking at the agent speaker rather than a human speaker. To prevent unreliable speech recognition in public exhibitions, the way Max (Kopp et al., 2005) used to acquire the inputs from the museum visitors is a keyboard, but this limits it to interact with the visitors one by one. It counts the number of multiple visitors standing in front of him by using skin color features but is not able to precisely track the visitors if they stand closely.

2.3 Content Management

The latest commercial image managers such as Adobe Systems' Photoshop Album (Adobe Systems Inc., 2004) and ACD Systems' ACDSee Photo Manager (ACD Systems, 2004) incorporate keyword searching and calendar view features. Thus, browsing image collections based on different viewpoints becomes possible. However, their search features only consider immediate usage and leave no cue for further information retrievals. Hence, they are not suitable for long-term management of personal memories.

PhotoMesa (Bederson, 2001) is an application for photo collection management with a zoomable user interface that provides the user a bird's-eye view of the managed photos. The zoomable interface improves the efficiency of image browsing in large collections; however, its automatic layout algorithm scatters the folders and makes it difficult to locate a particular folder or photograph in a large collection containing a large number of folders. The users cannot determine the folder locations; therefore, folders can be very difficult to locate when a lot of small folders are present. Moreover, PhotoMesa excludes semantic information associated with individual photographs, and therefore, the collection cannot be organized semantically. FotoFile (Kuchinsky et al., 1999) is a consumer multimedia organization and retrieval system that builds on the metaphor of an album to organize a personal multimedia repository; however, it lacks an overall view of all the contents and is restricted to a single hierarchy. PhotoTOC (Platt et al., 2002) uses the color histogram and timestamp information of digital photos to cluster personal photo collections into automatically generated event albums; however, it excludes the use of semantic information and lacks an overall view of the entire image collection.

Data Mountain (Robertson et al., 1998) allows its users to place bookmarks to websites on an inclined plane in a 3D virtual environment. It exploits spatial memory as a memory recall cue to improve the efficiency of information retrieval; however, its fixed surface size can only accommodate around 100 pages, and it cannot handle thousands of bookmarks simultaneously.

On the other hand, a number of researches similar to the KJ Method (Kawakita, 1975) have been proposed previously to address knowledge management or idea generation supported by spatial representation. For example, CAT1 (Sumi et al., 1997) and AA1 (Hori,

1994) utilize a spatial layout for ideas on a 2D surface to help their users generate new ideas. IdeaManager-iBox (Shibata & Hori, 2002) provides long-term storage of ideas, problems, and personal information. It supports the repeated refining of problems or ideas in daily life but does not utilize image information and spatial layout.

Chapter 3

Generic Embodied Conversational Agent Framework

This chapter proposes the basis of the other chapters of this dissertation, the Generic Embodied Conversational Agent (GECA) framework. The goal of this project is to provide a general purpose framework for developing ECAs as mentioned in chapter 1. It includes an integration platform, a set of communication libraries, high level protocols for ECA components as well as a reference starter toolkit that can be extended later for different applications. This chapter begins with an introduction of the requirements of a general purpose ECA development framework in section 3.1. Section 3.2 describes the design of GECA. Section 3.3 describes the essential GECA components including a character animator and a script language for describing human-agent interactions called GSML. Section 3.4 introduces the extensions to the minimum set of GECA components with a BDI (Belief-Desire-Intention) architecture. Section 3.5 ends this chapter with a comparison of ECA development with and without GECA.

3.1 The Requirements for a General Purpose ECA Development Framework

Like typical modeling of regular autonomous agents, an ECA needs to posses the following abilities:

1. Perceive verbal and non-verbal inputs from the user and the environment where the user is in.
2. Interpret the meaning of the inputs and deliberate appropriate verbal and non-verbal actions as the responses.
3. Perform those actions with an animated computer graphic character in a virtual environment.

In order to realize these abilities, various functionalities like sensor data acquiring, speech recognition, gesture recognition, natural language understanding, BDI planning, speech synthesizing, CG character animator and so on are required. Here, we call the modules that handle each individual function as *components* of the whole ECA system. In a 2002 workshop (Gratch et al., 2002), around 30 international ECA researchers already had intensive discussions about how to achieve a modular architecture and interface standards that will allow researchers in this area to reuse each other's work. However, this goal is still not yet realized except the work of SAIBA framework that is introduced in chapter 2. To achieve a common ECA component framework for general purposes, there are various requirements should be fulfilled and can be classified into three categories.

Integration Platform

A platform that can seamlessly integrate various ECA components and drive them to jointly behave as an integral ECA is indispensable. Such a platform has the following requirements.

Distributed and OS/programming language independence. Since the heterogeneous nature to utilize currently available software tools, components may be developed by various programming languages and run on various operating systems. Hence, the

ability for the integration framework to cover major operating systems and programing languages and allow the connected components to run on multiple machines is a necessity.

Modularity and reusability. This should be the heart of any integration approach. Component reusability can be maximized by cleanly divided functionalities of components and clearly defined interface between each other. Simpler functionalities handled by each component and lower interdependency improve modularity.

Support of various natural languages. As the advance of transportation, the world become smaller and smaller, the cross-culture issue has been emerging much more importance than before. However, due to the truth that western countries dominate the development of computer science field, the issues related to Asian languages or others are often ignored. To achieve generality of the whole framework, the flexibility to handle various languages need to be maintained.

Two-way communication among components. The ECA components do not only "pull" data from the others, but some of them such as sensor data processing components also have to "push" data to the others. Hence a mechanism which supports two-way data passing is required.

Real-time performance and timing control. Real-time response of the agent to user's inputs is one of the basic requirements of ECA systems. The latency of each part of the system needs to be kept as minimum while on-time execution of actions need to be guaranteed. Therefore, a strict temporal model is a necessity.

Ease the efforts to adopt legacy systems. Libraries and tools should be provided to ease the efforts to develop wrappers for adopting legacy systems to be connected to the architecture.

The Support of ECA Specific Functionalities

In contrary to general-purpose distributed architecture, for an architecture dedicated to the development of ECAs, the following supports are required.

Fusion of multi-modal inputs. In multi-modal interactive ECA systems, the relationship of user inputs from speech channel and other possible sensory channels needs to be identified correctly and trigger appropriate responses from the agent.

Synchronization between prerecorded tracks and run-time generated behaviors in outputs. Fixed length prerecorded tracks such as voice, music, or motion captured animation sequence need to be synchronized with variant length run-time generated animations.

Synchronization between verbal and nonverbal behaviors in outputs. Verbal and non-verbal behaviors are interrelated, supple each other and need to be synchronized.

Virtual environment control. Not only the virtual character itself but also the virtual environment that it lives need to be altered corresponding to the interactions of the agent and the human user, e.g. scene changes and camera manipulations.

User interruption. Provide the flexibility that allows smarter system to modify its current behaviors on-line instead of simply stops them and then launch the new ones.

A Reference Starter Toolkit

In order to show the usefulness of the new framework and to make it to be accepted easier by researchers, a reference implementation or a starter toolkit is necessary. The developers who are new to the framework can open the package and try the fully functioning example application immediately. The toolkit also should be able to be extended or customized easily in according to the researchers' needs. A minimum set of essential components can be considered.

User input acquiring component. There should be at least one component for the ECA to acquire the input from the human user. The input may be from a text console, a speech recognizer, a motion capture device or any other sensory devices.

Decision making component. There should be at least one component that decides agent's behaviors in responding to the input. It can be just a simple script engine or a set of components that form a complex deliberation process.

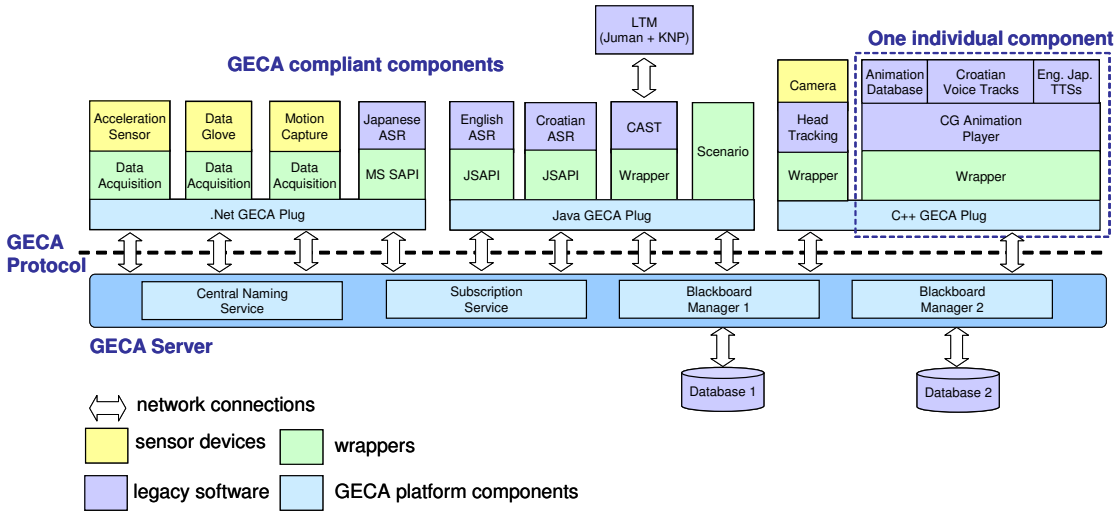

Figure 3.1: The conceptual diagram of GECA framework and the configuration of a multimodal tour guide agent

Character animation player. There should be a component that actually renders the agent's bahviors with character antimations.

3.2 The Design of GECA Framework

The GECA framework is composed of three parts, the integration backbone *GECA Platform*, communication libraries *GECA Plugs*, and a high level protocol *GECA Protocol*. Figure 3.1 shows the concept diagram of the GECA framework.

3.2.1 The Integration Platform of GECA Framework

ECA is not a new research area, and there are many excellent individual ECA systems like REA (Cassell et al., 1999) with various integration architectures have been proposed. However, contemporary ECA architectures are usually designed for specific applications, and their architectures typically feature fixed processing pipelines of functional components and thus can not be easily adapted to other applications. On the other hand, blackboard model is a methodology widely used in distributed and large-scale expert systems. Its basic idea is the use of a public shared memory where all knowledge sources read and write information.

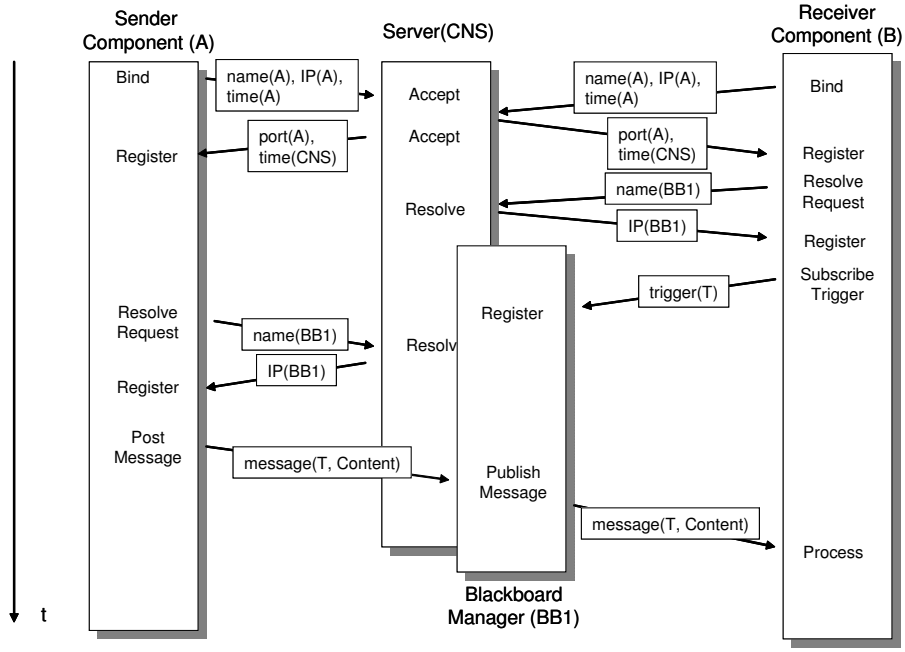

Figure 3.2: The message passing procedure on GECA Platform

The interdependency among the knowledge sources can be minimized, and thus it is considered suitable for integrating heterogeneous knowledge sources. Considering black board's convenience and generality in integrating various heterogeneous system components, we adopted it as the basic architecture of GECA Platform.

In GECA, multiple shared blackboards are allowed. There is a *GECA Server* providing simultaneously running threads for directory service (Central Naming Service, CNS), message subscription, and the managers of individual blackboard. The blackboard managers can be set up to use a MySQL (Sun Microsystems, 2004b) database for data storage. Components connecting to those blackboards share data with subscribe-publish message passing mechanism.

The procedure is described in Figure 3.2. When a component starts to run, it only knows the access information of GECA Server's CNS service. At first, it has to *bind* itself to the platform by telling the CNS service its unique name, IP address, and its local time. The CNS then establish a dedicated port and a dedicated connection for that component as well as its own local time. When the component receives these data from the server, it then register the data in its own cache.

The time information exchanged is used to synchronize the component's local time with the server. The synchronization method is the same as NTP (Network Time Protocol) (IETF, 1992, 1996) and can be described by Equation 3.1. T_s is the time when the server sent its reply, T_r is the time when the server received the client's query, t_s is the time when the component sent the request, t_r is the time when the component received the reply from the server, and θ is the compensation required to be applied on the component's local time. A dedicated synchronization message type is used for this purpose. Synchronization may be repeated for several times until it can not find difference from the server's time. The precision of this simple method depends on the granularity of the time management of the OS, the error is under 15.6 ms in MS Windows and is under 1 ms in Linux.

$$\theta = \frac{T_s + T_r}{2} - \frac{t_s + t_r}{2} \tag{3.1}$$

Then, if a component (receiver) expects to receive certain type of message, it first queries the CNS for the access information of the source blackboard. It then subscribes the message type (called trigger here) which it is interested in to the blackboard manager. It may also subscribe multiple triggers related to more than one blackboard manager to the subscription service at once (with a query to CNS at first). After this, every time when another component (sender) generates a message to the registered blackboard, the message will be forwarded to the components who registered this trigger by the manager. Every component can be a sender, a receiver, or both. To reduce the overhead of message forwarding, direct communication between components is allowed, too.

A simple and light traffic weight protocol, OpenAIR (mindmakers.org, 2005) is adopted as the low-level routing protocol for the communication among components, GECA server and blackboards. OpenAIR is a specification of XML message format for real-time interactive and distributed systems on a TCP/IP network. We considered that is suitable because its message format is very simple and it has some features like explicit timestamps.

The second part provided in the GECA framework is called GECA Plug libraries. They are extended OpenAIR Plug with GECA original classes and functions. Currently C#, C++ versions have been developed while the Java version is modified from the reference implementation. The purpose of the GECA Plugs is to absorb the differences caused by operation systems and programming languages and to make system development easier. By utilizing

GECA Plugs, an ECA developer only needs to implement a small wrapper for an existing software tool; then it can be plugged into the framework and cooperates with the other components. The third part of the GECA framework is the GECA Protocol; it is a specification of available message types and high-level XML message formats that are transferred on the GECA Platform. The detailed introduction of this protocol is left to section 3.2.2.

3.2.2 GECA Protocol

Based on the low-level communication platform of GECA framework, GECA Protocol (GECAP) is an XML based high-level communication protocol for the components. In GECAP, all data is represented as text and transferred by OpenAIR on the GECA platform. Every message has an unique *ID* as well as the slots of *type*, *posted timestamp*, *received timestamp*, *language*, and *content*. Each message type has a specified set of elements and attributes contained in the content slot. GECAP is a specification of message format style and a set of core message types, the syntax is not fixed and can be easily extended to meet the demands of individual applications.

Considering the information flow from the human user's inputs to the agent's responses and the system needs, GECAP message types can be divided into three categories: input phase, output phase, and system messages. Input and output phase messages can be further categorized into three layers, raw parameter, primitive action, and semantic interpretation in the sense of abstractness (Figure 3.3). Since components in GECA are connected as one level but not hierarchical, as shown in the figure, they can communicate with each other in mixed message layers. Components are not categorized into a certain layer, but each one of them can communicate with messages in multiple layers.

GECAP Message Types in Input Phase

The task of the components which generate input message types is to acquire and to interpret human users' inputs from verbal and non-verbal channels. The follows are some examples of defined input message types where "`input.perception.*`" types transfer primitive actions and "`input.raw.*`" types transfer raw parameters. Speech recognition result in type, "`input.perception.speech`," head movements such as nodding and

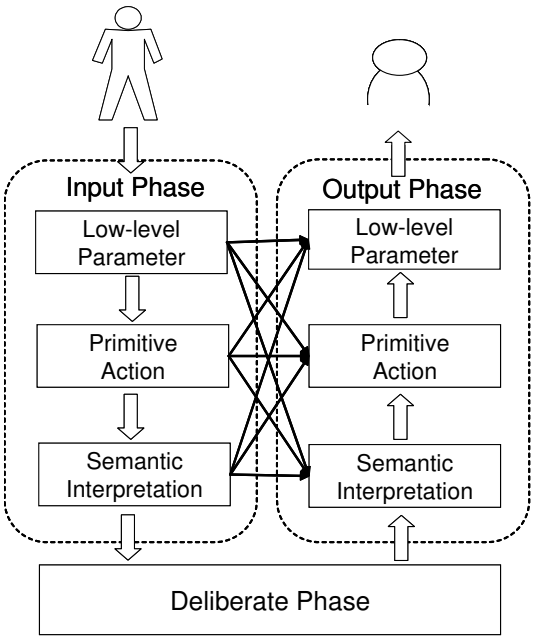

Figure 3.3: The layers of the abstractness of GECA messages

shaking that can be detected by an acceleration sensor and the results are sent in type, "`input.perception.head`," gaze direction that can be approximated by a head tracker as type, "`input.perception.gaze`," hand shapes acquired by data glove devices ("`input.perception.hand`"), the angles of the arm joints that can be approximated by three motion capture sensors attached on each arm ("`input.perception.arm`"), predefined hand gestures which is recognized by motion capturing devices are transferred in type, "`input.perception.gesture`," convenient pointing gesture which can be detected by a motion capturer or even a mouse ("`input.perception.point`").

The following XML segment is an example of an "`input.perception.speech`" type message. This message type also utilizes the language attribute of content slot of OpenAIR to store the recognized natural language with values like "`English`."

Listing 3.1: An XML segment that represents a speech recognition result

```
1 <Perception Begin="1175083369171" Duration="500" Weight="1.0">
2     <Hypothesis Confidence="0.9">what is this</Hypothesis>
3     <Hypothesis Confidence="0.1">what is these</Hypothesis>
4 </Perception>
```

The recognized result is contained as plain text in the `Hypothesis` element. Programs like speech recognizer or gesture recognizer usually have ambiguity in recognizing the data from real world sensors. The `Hypothesis` elements are used to present a list of hypotheses of the recognition result on a single input event with confidence ratings in values from 0 to 1. `Begin` attribute stores when this input event begins with the absolute time represented in milliseconds while `Duration` attribute stores how long the input event lasted. The following XML segment is an example of an "`input.perception.point`" type message that represents a position on the 2D screen where the user is pointing by performing a pointing gesture or by using a pointing device:

Listing 3.2: An XML segment that represents a pointed position

```
1 <Perception Begin="1175079954578" Duration="2000" Weight="0.5">
2     <Hypothesis Confidence="1.0"><Point X="0.2" Y="0.3"/></Hypothesis>
3 </Perception>
```

GECAP Message Types in Output Phase

The only actuator of software based ECAs is the character animation player. This player plays plain text with TTS and drive the CG character to move in the virtual environment when a command message arrives in real-time. Although current prototype GECA player is implemented by using commercial software, visage|SDK (Visage Technologies AB, 2008), the design of GECA's output message format is not dedicated to Visage and should be able to be ported to other animation systems. The player is described detailedly in section 3.3.1.

System Message Types

There are system controlling message types such as "`system.status.player`" or "`system.control.player`" to query the status of the ECA character (e.g. whether the character is playing an animation or idle) or make the character to stop speaking and playing any animation, etc.

3.2.3 Implementation

We have completed the first development of the GECA server. It is implemented in Java and the backboard is implemented on regular relational databases (MySQL). It becomes rather stable so that we can add new components running on multiple computers and are connected to the GECA server through C#, C++ or Java GECA Plugs. So far, we have implemented several ECA systems for different applications, by introducing GECA components such as Japanese spontaneous gesture generator (Y. Nakano et al., 2004), head tracker (Oka & Sato, 2005), hand shape recognizer, nodding detector, scenario interpreter, speech recognizer and the CG animator.

3.2.4 Real-time Capability of GECA Platform

In order to achieve real-time human-agent interactions, the GECA platform has to transport all components' data without noticeable delay. However, the data transfer rate highly depends on computer hardware, network environment, message size and component topology, it is not reasonable to conduct a strict evaluation on whether GECA performs fast enough or not. Instead of that, we measured GECA platform's performance characteristics to analyze in what circumstances sufficient performance can be achieved. From the point of view of applying network platform in real-time interactive systems, there are three main concerns.

Bandwidth: can the throughput of the platform fulfill the requirements of bandwidth starved components?

Latency: is the delay of network transfer noticeable for real-time applications?

Scalability: does the performance decay dramatically when the system scales up (more components connected)?

As the reference of typical data transferred on GECA, Table 3.1 shows the data rate (binary) of the data streams of motion capture devices which are considered to have relatively higher requirements on bandwidth. MAC 3D (MotionAnalysis Inc., 2009), PhaseSpace (PhaseSpace Inc., 2009), and MotionStar (Ascension Tech., 2009) are the available motion capture devices in our laboratory. The number of data per frame presented is the number of

Table 3.1: Bandwidth requirement of motion capture devices. Data size is represented in bytes

	data size	# of data/frame	frame size	Max. fps	Max. KB/s
MAC 3D	12	15	180	300	54.0
PhaseSpace	16	24	384	480	**184.3**
MotionStar	24	8	192	120	23.0

sensors/marks attached on the upper body of each participant in our data gathering experiments. Maximum frame rate denoted in this table is the capability of the sensor hardware which is not necessary for all applications. The higher the frame rate the more detailed body movements can be recorded, but usually a frame rate at 30fps is enough for most applications. From this table, the device requiring highest bandwidth is PhaseSpace (184.3KB/s) at its highest frame rate (480fps). The data rate will increase to 276.5KC/s (or 17.3KC/s at 30fps, C denotes character) if it is sent via plain text in regular Base64 encoding (IETF, 2003).

The time since one component (sender) starts to send a message to the server until the time when another component (receiver) receive that message from the server is measured. The measurement is done with two PCs connected with 1Gbps closed LAN. In order to get precise time measurement, simulated components are running on the same PC. The one running GECA server is equipped with a four-core 2.8GHz CPU, and the one running simulated components is equipped with a two-core 2.9GHz CPU. Various sizes of user data, number of sender-receiver pairs (one stream) are measured 200 runs per stream.

Bandwidth

Figure 3.4 shows the average data rate of each stream. From the measured data, it can be observed that the data rate tends to decrease when the number of stream or the message size increases. Moreover, the data rate is less than $1/n$ (n = the number of concurrent streams) of that when there is only one stream in large message settings. From the data, the bandwidth of GECA server has spare room when there are two maximum-frame-rate (553KC/s) PhaseSpace data sources (e.g. two participants) connected. If the data is transferred in binary

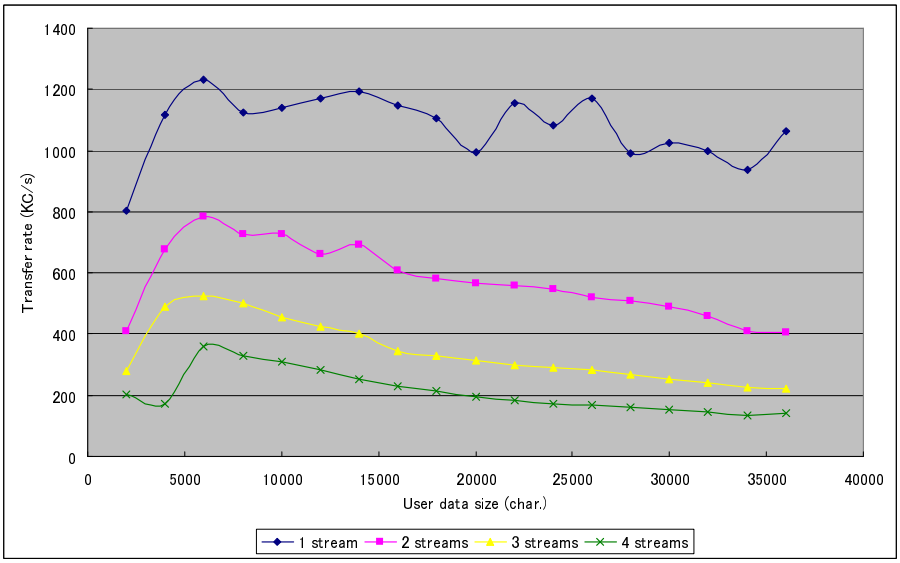

Figure 3.4: The relationship between average transfer rate of each stream and the number of concurrent streams

mode, four sources are allowed. If the frame rate is lower, e.g. 30 fps, then dozens of data sources (participants) are allowed.

Latency

According to the ITU-T recommendation on QoS (quality of service) (ITU-T, 2002) of network applications, the tolerance of delay ranges from 100 ms (highly interactive systems) to 400 ms (interactive systems). Figure 3.5 shows the data transfer time on GECA platform with different settings of message size and the number of concurrent streams. As shown in Table 3.1, the typical GECA messages usually only carry several hundred characters of user data, but in order to observe the tendency of the network tendency, huge message sizes are tested as well. From the measured data, the latency increases with message size and the number of concurrent streams but is close to 0 ms when message size is small (less than 5,000 characters). Therefore, it can be expected that if the message size is small and is transferring an event, the delay should not be noticeable as long as there are no many concurrent streams. Note that this transfer time is for one individual message, it can be several times longer in practical if that perception or action the agent involves more

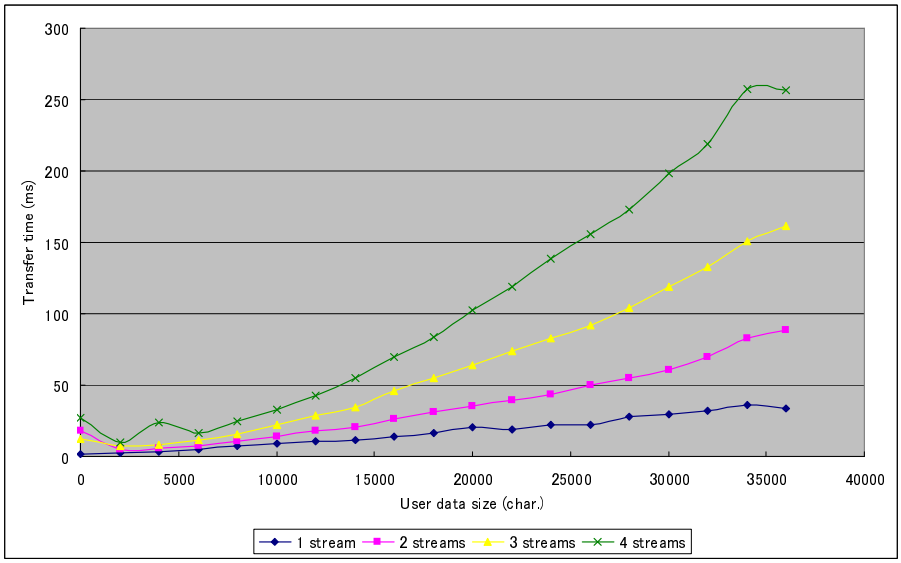

Figure 3.5: The relationship between average data transfer time of one message and the number of concurrent streams

components.

Scalability

Figure 3.6 shows the relationship between total data transfer rate and the number of concurrent streams. The total transfer rate is supposed to depend on network specification heavily and should vary system to system. In the experiment setting, it is always around 1M C/s but gradually decreases when the size of message or the number of concurrent streams increases. From the measurement results, the system performance was fairly stable and dramatic decay was not observed even in extreme settings.

3.3 Essential GECA Components

This section describes the reference implementation of two essential components of GECA framework. One is a 3D CG character animator and the other one is a script language for describing human-agent interactions with its executor. They can latter be modified according to different system needs.

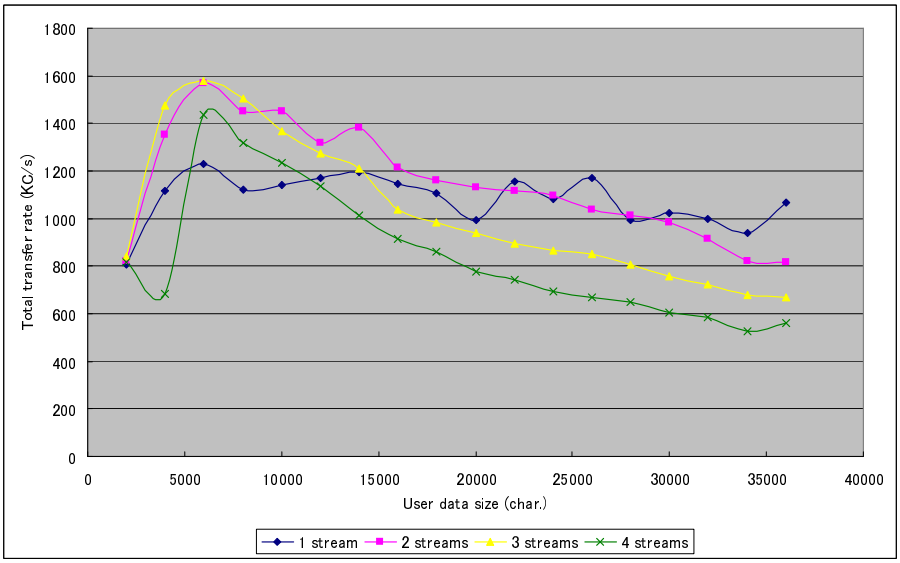

Figure 3.6: The relationship between average total data transfer rate and the number of concurrent streams

3.3.1 GECA Capable Character Animation Player

Current GECA character animator is developed with Visage|SDK what is a MPEG-4 FBAP compliant CG character rendering library. All parts of the full 3D anthropomorphic character like the limbs, fingers, eyes, mouth, and so on can be programmed to perform arbitrary animations that can be done by a real human. The animation player provides the support of Microsoft SAPI (Speech API) (Microsoft Corp., 2001) compatible TTS engines for the character's speech with synchronized lip animations. To simplify the problem and also because a picture usually looks more realistic than a full 3D environment which lacks enough details, the virtual environment for the agent's activities is represented by switching 2D background pictures.

The XML segment in Listing 3.3 is an example of the content of the message in type, "`output.player.utterance`" that is interpreted and executed by the animation player. Each message of this type contains a trunk of animation descriptions in an `Utterance` element. Because the paralleled running architecture of GECA, more than one component may communicate with the player simultaneously. The arrived multi-modal utterances are then

stored in an ordered queue in the player for further animation playing. When the player finished playing an utterance, it sends a feedback message in type "`output.player.result`" with the utterance's reference ID to indicate whether it is successfully *done*, *interrupted*, or *ignored*. `Sentence` element is the basic unit that will be executed one by one by the player. If the player is interrupted, it continues currently running sentence until it finishes. This utterance is then interrupted and the other buffered ones are then ignored.

Nonverbal behaviors of the ECA are described in the `Action` elements, and their synchronization timing information is encoded by the containing relationship with the verbal `Phrase` elements. The timing to start to play the specified animation is determined by the position of the opening tag relative to the verbal utterance. In the case where the agent will not say anything, a `Delay` attribute is used to specify when the animation will be played relative to the beginning of the template. The playing of this animation will end when the agent speaks to the closing tag of `Action` element or meets the time specified by the `Duration` attribute. `Sync` attribute specifies the temporal relationship between the actions in an utterance. There are three possible values: "`WithNext`", "`BeforeNext`", and "`PauseSpeaking`" stands for do not wait for this action, to wait for this action to end and to pause TTS while executing this action respectively. Figure 3.7 depicts how the animation described by this code segment will be rendered by the player.

Listing 3.3: An XML segment that represents a multi-modal utterance for the player

```
<Utterance>
  <Sentence>
    <Phrase>Hello.</Phrase>
  </Sentence>
  <Sentence>
    <Action Type="expression" SubType="smile" Duration="2300" Intensity="0">
    <Action Type="bow" Duration="1700" Intensity="1" Sync="BeforeNext"/>
    <Phrase>My name is Dubravka and I will</Phrase>
    <Action Type="beat" SubType="d" Duration="600"/>
      <Phrase>be your tour guide agent of Dubrovnik city .</Phrase>
    </Action>
  </Sentence>
</Utterance>
```

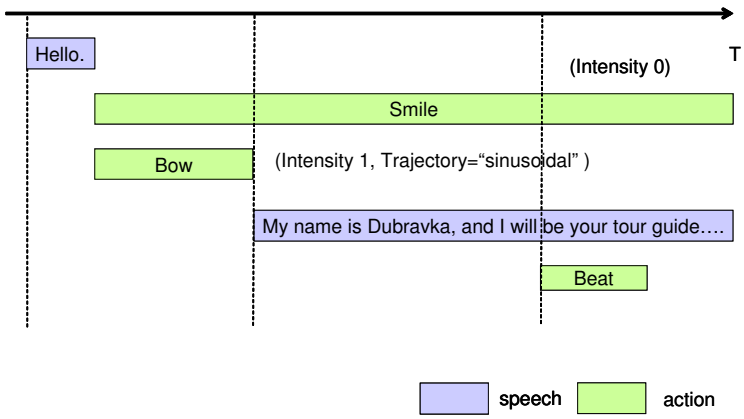

Figure 3.7: Synchronization of speech and nonverbal animations in the player

A set of attributes are defined to complement the character animation description in `Action` element. `Subtype` specifies another action in the same category if available. `Intensity` specifies the strength of the animation if specifiable e.g. how much the character bows or how much it smiles. `X`, `Y`, and `Z` specify a position in the virtual world if the action has a target or destination in the virtual space, e.g. walking, pointing, gazing actions. `Direction` specifies a direction of the action if available, e.g. in which direction the character should face after a walking animation. In current implementation of the player, routine generated animations are modeled in three phases including *attack*, *stroke*, and "decay." Each animation starts from the neutral position of the character, the joints of the character are rotated during the attack phase until they reached to the destination position (stroke). The angles of the joints are kept the same during the stroke phase for a while and then decays to the neutral status again (Figure 3.8). The time length of attack and decay phase are modeled as the same. `Trajectory` specifies the temporal function to change joint parameter values during the attack and decay phases. "`Linear`," "`Sinusoidal`," and "`Oscillation`" are currently available values. TTS engines' prosody information specifying tags are not a part of GECAP but they are allowed to be inserted into `Phrase` elements. GECA components will ignore them and pass them to be processed by the TTS.

Since there is no reasonable boundary for possible actions that can be done by a human or an ECA character, we are not going to specify a full set of the actions but only defined the syntax to specify the animations and a set of animations that are supposed to be most

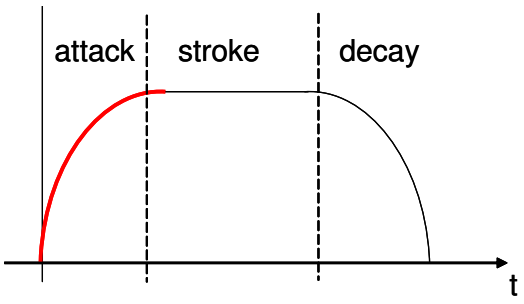

Figure 3.8: The trajectory of nonverbal behavior animations

frequently used. The set of available animations should be application dependent.

A special action type created is the `PlayTrack` action, this action plays a fixed length media like a background music, voice, animation sequences modeled in 3D modeling software (currently only 3ds Max (Autodesk, 2009) is supported), or even human movements recorded by a motion capture device. It then can be used to implement an ECA system in a language which has no available TTS engines. For example, an agent speaking Croatian can be implemented with pre-recorded human voice tracks and lip actions. The `Delay` attribute can be utilized in this case to synchronize the tracks with each other. The four types of animations, *routine generated animations synchronized with TTS*, *independent routine generated animations*, *track synchronized with TTS*, are *independent tracks* are running as parallel threads and are synchronized with each other by the player.

This syntax provides the distinguishing features include word-level precisely aligned non-verbal behaviors on-the-fly as well as multi-language support. Many TTS engines (e.g. the popular MS SAPI compliant ones) can not provide timing information in prior. Character animation description language like BML that requires timing information to schedule animations can only work with limited number of TTS engines.

Since the internal presentation of the character's animation is the standard MPEG-4 face body animation parameters, in the case of the reference implementation of the player, the raw parameters is thus MPEG-4 FBA parameters. Message type "`output.raw.FBAP`" is defined to carry the used parameters' numeric value and drive the character in real-time. Figure 3.9 shows an example system where the user avatar and the computer controlled agents are driven in real-time by "`input.raw.arm`" and "`output.raw.FBAP`" messages. The avatar replays the user's hand gestures such as beckoning while ten computer controlled

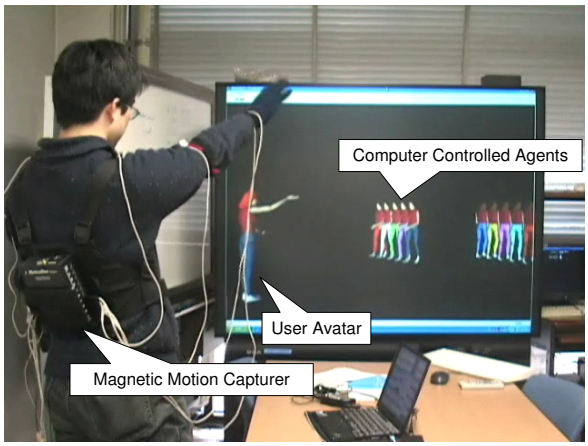

Figure 3.9: A cultural difference experiencing application with one user avatar and 10 computer controlled agents driven by raw parameters to raw parameters

agents react to those gestures pretending that they are Japanese or British. The user's actions are captured by a magnetic motion capturing device and interpreted to low-level joint angles to drive the avatar character in real-time. The computer controlled agents are driven by individual reflexive controlling components and a common BAP catalog component. They are driven by low-level MPEG-4 BAPs in real-time, too.

3.3.2 GECA Scenario Mark-up Language (GSML)

To achieve really natural conversation between the ECA and a human user, many factors need to be considered in the deliberate process of an ECA: natural language understanding, inference engine, knowledge representation, dialogue management, personality and emotion model, social role model, natural language generation and so on are required. Considering the complexity and the fact that the present level of technology is still impossible to drive an ECA to behave like a human in an indistinguishable level, instead of a block of complex deliberate process, we have defined a script language, GECA Scenario Mark-up Language (GSML) that describes the interactions between the user and the agent as the basic implementation of the deliberate process of a GECA agent. A script definable ECA is less general than a deliberative process, but it will be much easier to create contents and should be useful enough for simpler ECA interface applications.

GSML shares the most basic concept of AIML (Artificial Intelligence Markup Language) (A.L.I.C.E. AI Fnd., 2005) which is a widely used script language for defining text based chatbot agents on the Web. An AIML script represents an agent's knowledge that is composed by a set of `Category` elements. One `Category` contains a pair of `Pattern` and `Template` that describes one of the possible conversations between a chatbot and its human user. When there is a user's utterance comes into the interpreter, that utterance is matched with all of the defined patterns, the agent then responses with the utterance described in the corresponding `Template` element. However, AIML can not be applied to the ECA context due to the following reasons: supports English only, unexpected template may be triggered because the same patterns can not be distinguished in different circumstances, can not describe non-verbal behaviors of neither human user nor agent, no way to specify objects in the virtual world, agent behaviors need to be triggered from the human side.

GSML extends AIML's syntaxes to cover more complex situations in face-to-face conversations in an ECA setting. The complete document type definition (DTD) and reference of GSML is listed in Appendix A. Extending AIML's one-layer categories, GSML represents the human-ECA conversations as states and the transitions among them. Figure 3.10 shows the additional three layers of the hierarchy of GSML categories. In GSML, one `Scenario` defines an interactive scenario between the ECA and the human user. A scenario can contain one or more `Scene` elements while each `Scene` means a physical location in the virtual world and is coupled with a background image. In an individual, there may be one or more conversational `State` elements. Each `State` contains one or more `Category` elements. The conversational states are linked by `Transition` specifications described in `Template` elements. Further, templates can be triggered right away when conversational state transition occurs without user inputs. The Scenario-Scene-State-Category hierarchy narrows the range of possible categories into a conversational state and prevents the problem that templates may be triggered unexpectedly in AIML agent which practically has only one conversational state. Besides, the `Language` attribute in states allows a multi-lingual ECA to be defined in a single GSML script.

GSML's patterns and templates do not only present verbal utterance of the agent but are also extended to describe non-verbal behaviors of the agent and the human user. `Action` tags that specify face or body animations can be inserted into the utterances of the agent,

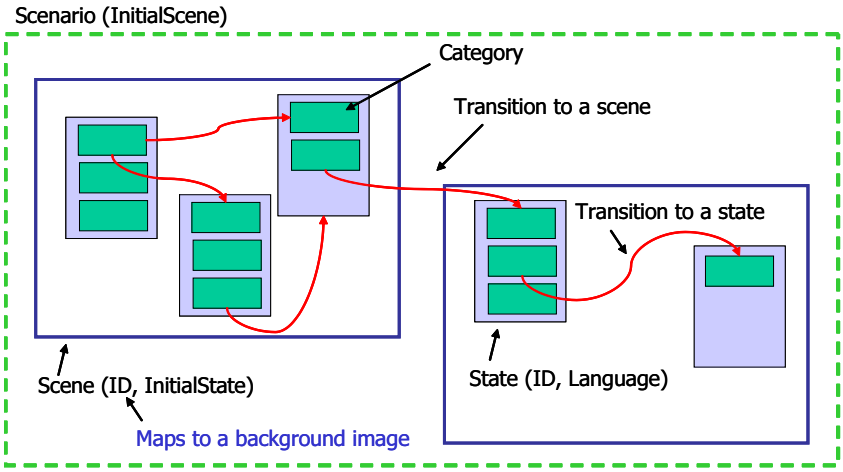

Figure 3.10: The diagram showing the relationship between`Scenario`, `Scene`, `State`, and `Category` elements in GSML

the timing information is specified by the position of the `Action` tags in the utterance texts. The action tags (`Speech`, `Point`, etc) can be inserted inside the `Pattern` tags then the corresponding template will be triggered if the user does that non-verbal behavior. Further, areas of the background image can be named by `Object` elements and can be referenced (e.g. pointed at or gazed at) by the user during the multi-modal interactions.

By observing usual face-to-face communications between humans, we can find non-verbal behaviors are the indispensable counterpart of verbal utterances. For example, the verbal utterance "What is this?" with a pointing gesture is a very typical example. Without the pointing gesture, which object that this "this" is mentioning becomes ambiguous. On the other hand, a pointing gesture can not fully convey the user's intention, either. Generally, the order, combination, and occurrence of multi-modal perceptions and their relationship are difficult to be described and identified. Like the discussion in the specification of W3C's multi-modal interface description language for Web browsing, EMMA (Extensible Multi-Modal Annotation markup language) (W3C, 2004), it is not easy to implement a general algorithm for multi-modality fusion. In GSML and its interpreter (the scenario component), we adopted a simplified description for multi-modal perception of the ECA and a relatively simple mechanism to solve reference ambiguities. Since EMMA is designed for similar purpose as GECAP's input phase and GSML, some of the element names that we are using

are inspired from those defined in EMMA, however, what do they mean and how they are used are very different to those in EMMA.

`Set` element means a non-ordered set of multiple verbal or non-verbal perceptions and every one of them must be fulfilled. `OneOf` element means at least one of the multi-modal perceptions needs to be fulfilled. `Sequence` means the multi-modal perceptions need to be performed by the human in the specified order. The three specifiers can be further nested with each other. Whether two multi-modal perceptions occur concurrently is judged by the period coverage of involved perceptions according to the `Begin` and `Duration` attributes in the message sent from the sensor data acquiring components. The scenario component keeps a current status of the multi-modal perceptions and triggers the corresponding `Template` if any one of the available patterns defined in the current conversational state can be exactly matched. This matching is calculated every time when a new input message arrives. The combination which has highest value of the sum of the product of confidence and component weight is chosen in the matching (Figure 3.11). Listing 3.4 is an example code segment describing the interaction between the human user and a tour guide agent.

Listing 3.4: A segment of a GSML script

```
<Scene ID="Entrance" InitialState="Greet" X="1250" Y="937">
  <Objects>
    <Object ID="Fountain" X="900" Y="0" Width="350" Height="937"/>
    <Object ID="Monastery" X="0" Y="0" Width="377" Height="937"/>
  </Objects>
  <State ID="Greet" Language="English">
    <Category>
      <Pattern>
        <Speech>hello</Speech>
      </Pattern>
      <Template>Hello, my name is Dubravka, and I am the guide here. Where do you want to go?
        <Action Type="pointing" Duration="1000" Direction="right">The fountain</Action>or
        <Action Type="pointing" Duration="1000" Direction="left">the monastery?</Action>
      </Template>
    </Category>
    <Category>
      <Pattern>
        <OneOf>
```

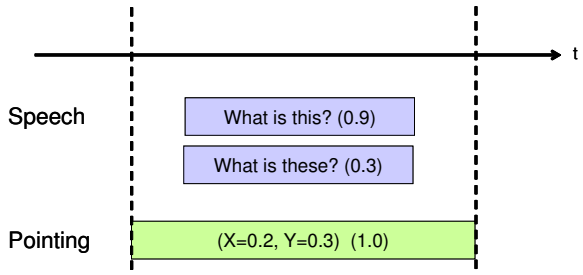

Figure 3.11: Multi-modal fusion in GSML

```
19          <Speech>fountain</Speech>
20          <Set>
21            <Speech>I want to go there</Speech>
22             <Point Object="Fountain"/>
23          </Set>
24        </OneOf>
25      </Pattern>
26      <Template>Please follow me here.
27        <Transition ToScene="Fountain">
28      </Template>
29    </Category> .....
```

The fore part of this code in Listing 3.4 specifies the scene with a background image that can be identified by the scene id, "`Entrance`." The `Object` elements specify two areas of the background image, "`Fountain`" and "`Monastery`." These areas are used to in the matching of the coordinates sent from some pointing component with the `Object` specifiers in second `Category`. According to the description of perception specifiers, when either one of the two circumstances is fulfilled, a conversational state transition to the initial state of the scene, "`Fountain`" will be triggered. When the human user says "fountain", or when the user says, "I want to go there" while performing a pointing gesture on the screen where the position is recognized as an `X` value from 0.72 to 1.0 and a `Y` value from 0 to 1.0 at the same time.

A `Template` is transferred to the player as an `Utterance` element in GECAP. A well implemented TTS engine adjusts its intonation output in the unit of sentences rather than just speak out words. In order to fully take advantage of this feature, the agent's utterances

are broken into sentences according to punctuation marks. The contents of a template is then broken into phrases and sentences as described in section 3.3.1. The sentences are then enclosed with `Sentence` and `Utterance` elements before they are sent to the player or the other components.

3.3.3 Extended GSML for Rule Based Agents

Comparing to previous system which merely matches recognized speech inputs and non-verbal inputs with predefined patterns, a variable system is introduced. Following information state theory (Traum et al., 1999; Larsson et al., n.d.), the interaction between the agent and one of two users are described by a set of variables like a snapshot. 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 arability of the users, `UserAttention` represents how much the users are paying attention to the system, `Addressee` specifies whom should be the addressee of agent's next utterance, etc.

The values of these variables are updated with the agent system's internal status, perception events sent from the speech recognition components and non-verbal input interpretation component. How the value of the variables should be updated can also be specified by the script designer in the script as the effects of particular input patterns. `Effect` element is introduced into Template element for this purpose. An input event can cause the values of particular variables to be bound to, added with, or subtracted with certain values.

The syntax of the patterns defined in GSML scripts is also extended. `Predicate` element is introduced to represent a test on the values of a variable. The value of the variables can be tested to be equals to, less or larger than certain values.

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 via the perceptions from outside world or the users and picks first valid template which made all of the conditions (predicates) true to perform. Therefore, the rules like the tour guide agent should walk to the front to greet when there are users presenting in the user area, say goodbye to the user and go back to the initial position when the user left the user area and so on

can be specified in the script.

`States` limit possible patterns that will be used in matching in current conversation situation and thus isolates the interference from other states which may happen to have the same triggering patterns. Because of the lacking of context management mechanism in the agent's behavior control, there is no way to justify whether a user answer is related to last question asked by the agent. However, for example, when the agent is going to ask a yes/no question like "Do you need a tour guide?", a transition to a specific state representing the question can isolate the question under discussion from the other yes/no questions.

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

Unlike a full dialogue managing central component, the disadvantage of this approach is: the agent does not conduct a plan that contains multiple steps to achieve certain goal. The agent's behaviors are driven by the events occurred in outside world. The management mechanism of information like grounding or topics is not included in the script execution kernel. These features are still implementable via the manipulation on but are left as script programmer's responsibility. The extended GSML is evaluated with the procedure of Algorithm 1, 2, and 3.

3.4 GECA in a Belief-Desire-Intention Configuration

BDI (Belief-Desire-Intention) model is a classic architecture of the deliberate process of autonomous agents. Instead of the reactive behaviors of a chatbot or GSML programmed agents, a BDI agent has the goals what it wants to achieve (desire). Consequently, it deliberates the rational actions which are supposed to lead to the goals (intention) depending on current understood status of the world and itself (belief). BDI architecture is therefore more suitable in the applications where the agent needs to complete certain tasks. We are also developing a BDI agent based on GECA framework in a sophisticated instrument instructing task where the agent tries to achieve the goal that the user understands its instructions (Hacker et al., 2009). This application is composed as the architecture which is similar to

Algorithm 1 The main loop of the evaluation process of extended GSML scripts

```
while bShouldRun do
  if bInterrupted then
    clear PendingTemplates
  end if
  for all category c of GlobalState do
    if match(pattern p of c, InformationState) and c is not expired then
      execute(template t of c)
    end if
  end for
  for all category c of CurrentState do
    if hold(pattern p of c, InformationState) then
      execute(template t of c)
    end if
  end for
end while
```

Algorithm 2 The *match* routine for the evaluation of a `Pattern`

```
for all predicate p of pattern P do
  if compute the function of p with the data from InformationState ≠ true then
    return false
  end if
  if speech = the text of P then
    return true
  end if
end for
```

Algorithm 3 The *execute* routine for the evaluation of a `Template`

```
send the multi-modal utterance of t to the player
PendingTemplate ⇐ t
if the animation of t is done by the player then
  PendingTemplate ⇒ t
  for all effect e of t do
    InformationState ⇐ the computation of the function of e
  end for
end if
```

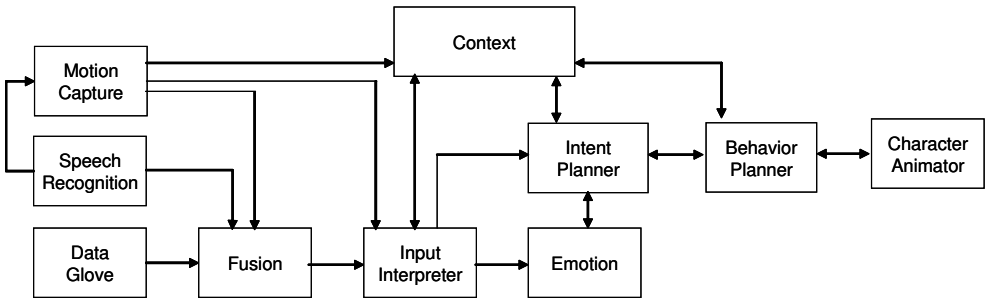

Figure 3.12: A BDI configuration based on GECA framework

SAIBA framework at its output phase and is shown in Figure 3.12).

The human user's behaviors are acquired by the *Motion Capture*, *Speech Recognition*, and *Data Glove* sensory components. By the combination of the data from motion capture and data glove, the pointing gesture, the head movements, and the posture of the user can be detected. Coherent modalities forming user behavior are modeled allowing for the different modalities to be recognized independently from each other by representing them as units according to their necessity of coherence. For example, the utterance of pointing out an object on the screen and additionally expressing which type of object the user is pointing to by speech consists of three modality types (hand, arm, and speech).

Contextual behavior interpretation is necessary to create a more natural conversational situation by enabling the agent to react more appropriately to the user's behavior. Given that not all user behaviors have the same intentions throughout the whole discourse the *Input Interpreter* component needs to assign an intention to any of the user's behaviors by considering the current discourse context which is frequently updated by the intention planning and emotional behavior realization unit. Raising a hand in the beginning of a conversation will most likely indicate a greeting while the same gesture might indicate the agent an interruption of its discourse in a different context. This component is therefore connected to the *Context* component which contains a current representation of the virtual world in simple spatial and semantic terms.

The Context component stores all the relevant data for the initialization and processing of our system. It describes the object, their sizes and coordinates and meta information about the objects relations to the environment and other objects. The object's coordinates are essential to a successfully identify the aim of the pointing gesture carried out by the user.

The meta information is used to describe spatial associations to other parts of the environment with relative terms (“above,” “behind,” “to the right,” etc.) and visual characteristics (“big,” “red,” “round,” etc.). The meta information is used to provide an identification of objects by natural speech input. Furthermore, it contains information about the objects including general description, maintaining advices, error handling, and so on. The Context component also stores data needed for generating intention and behaviors. For example, personal and interpersonal data, like names, genders, relationship, language, etc. Internal states of the *Emotion* component (e.g. user interest) are stored in the Context, too. Besides, there are dictionaries within the Context which are used by the *Behavior Planning* for the generation of basic behaviors of the agent.

The Emotion component administrates the emotional states of the agent and the user. The internal change of the emotions is realized with an affective behavior using an approach of describing emotions proposed by (Becker et al., 2007). The description of emotions is realized by notation three values for pleasure, arousal and dominance. Additional states (e.g. user interest in current conversation) for describing the suggested feelings of the user are managed by this component. If these stages reach critical levels, the Emotion component triggers an event that is processed by the *Intent Planner* and may result in a change of the plan.

The planning of the next intentions is realized with the BDI model. In this system, the Jadex (Pokahr et al., 2005) BDI engine was adopted. During run-time, the agent is able to load his beliefs out of the context and can update them after processing an input. The Intent Planner component uses the output provided by the Input Interpreter and the situation’s context to generate intentions. It is possible that the Intent Planning receives events by the Emotion component (e.g. emotion state becomes critical). In this case, this event is integrated in the process of generating a new intention.

The *Behavior Planner* generates animation descriptions in the same way as the GSML interpreter. Intentions sent from the Intent Planner provide the input for this component’s planning. By using different dictionaries the communicative tag of this description is substituted to multi-modal utterances for the Character Animator component. The dictionaries are chosen on base of the language attribute. Within the dictionaries are building blocks for behaviors including vocal output and animations. The combinations of these behaviors differ

in the dictionaries depending on the language and associated culture. Therefore, a "greet" may be substituted by "Hello" and waving hand for English language in contrast to "Konnichiwa" and taking a bow for Japanese language. Additional information (e.g. emotion, personal and interpersonal information) are used for modifying the way how the behavior are realized (e.g. speed and pitch of verbal output, smiling or serious facial expressions).

3.5 Expected Advantages and the Disadvantages to Use GECA

Comparing to previous architectures, GECA framework is expected to have the following advantages:

- Components developed with different programming languages and running on different OS's can be integrated easily.
- Components which require heavy computation can be distributed to multiple computers and improve overall system performance.
- The single-layer component hierarchy shortens the path of decision making and eases the support of reactive behaviors.
- Explicit temporal information and synchronization specifiers ensures that components are synchronized.
- ECA systems with various features can be configured easily with different component topologies.
- The weak inter-dependency among the components allows on-line switching of components and on-line system upgrading.
- Direct component communication and the multiple blackboard capability can lower message transmission loads.
- The loose modular architecture eases the collaboration between multiple developers and encourages the sharing of research results.

Nevertheless, comparing to a dedicated architecture, developing an ECA system with GECA may face the following disadvantages:

- The topology of the components may not be optimal in the sense of the length of links, the system design may be more complex (more components, more links) in some cases.
- The performance penalty due to heavier reliance on the network than a dedicated design where the functionalities of an ECA can be packed into fewer and larger components that can run on the same machine.
- Available components may not perfectly match the requirements of the project and not easy to extend.

GECA is not designed for building most advanced ECA or best performance but for easing the development efforts in building various ECA systems. It has its advantages and disadvantages, the developers are required to make a choose upon whether GECA is suitable to their projects.

3.6 Conclusions

This chapter represented the Generic Embodied Conversational Agent (GECA) framework that covers the information process from the detection of the human users to the behavior outputs of the ECA. A script language (GSML) that specifies an ECA's behavior is also introduced. Three example systems for preliminary evaluations are also introduced. The goal of this project is to make the framework publicly available with a reference ECA toolkit which can be used to build ECA systems in instant and can be extended easily.

We found the following problems in developing this framework. The description on the multi-modal input from the user is still quite trivial and can only capture simple actions done by the human user. We would like to strengthen this part to capture more complex conversational circumstances in the future. It was difficult to develop general purpose components for various applications, for example, to show subtext in the animator. Sometimes, there was problem in timing because we can not get direct control inside a model, for example,

the TTS engine starts slowly in first run trial. The available action set is still small and can only be used with limited applications.

Chapter 4

Developing a Tour Guide Agent with GECA

Follow the detailed description of the GECA framework in last chapter, the natural concern is whether GECA can be used to build a nontrivial ECA system and how to use GECA to build it. This chapter describes an ECA named Dubravka as an example application of GECA framework. The Dubravka agent was developed in an ongoing international collaborative project aiming to build a tour guide agent who is adaptive to users from general Western, Japanese, and Croatian cultures. The purpose of this project is not to pursue technical breakthrough but is to show the usefulness of GECA framework and to provide a testbed for it. At first, we show that the fully functional ECA, Dubravka can be created at low-cost in a relatively short period by using GECA. The main part of this chapter has a detailed description of the implementation of Dubravka. At last, we investigated the issues emerged in the situation where there are more than one users.

This chapter is organized as the follows, section 4.1 introduces the objectives of this project and the four-week eNTERFACE'06 workshop where Dubravka was created. Section 4.2 describes the cultural issues involved in the Dubravka agent project. Section 4.3 describes the implementation details of building the Dubravka agent. Section 4.4 describes potential extensions to the Dubravka agent with pluggable culture modules. Section 4.5 describes the extension of the Dubravka agent in eNTERFACE'08 workshop where we investigated multiple-user issues. Section 4.6 concludes this chapter.

4.1 The eNTERFACE'06 Workshop Project

This study was started during the eNTERFACE'06 workshop that focused on the topic of multi-modal human-computer interfaces and was held in Dubrovnik, Croatia in the summer of 2006. Contrary to regular workshops where the researchers only present their research results but do not actually work, the principle of this workshop is to invite volunteer student participants to collaboratively develop proposed research projects in a relatively short four-week period and then present their research results.

The title of our proposed project was "An Agent-Based Multicultural User Interface in a Customer Service Application." After the announcement of the project proposal in sponsoring universities, we got five student members in our team where two of them did not belong to our research group. On the basis of the discussions among team members prior to the workshop, the target application was decided to be a tour guide agent for Dubrovnik city. The entire old town of Dubrovnik has been designated a UNESCO World Heritage Site. Dubrovnik is a famous sightseeing spot and attracts thousands of tourists from all over the world, especially in summer because of the attractive festivals in this period. Since most of the team members come from Japan or Croatia, it was most convenient to gather first-hand Japanese and Croatian cultural information, where the differences are supposed to be fairly obvious. The agent was given a young female appearance and was named Dubravka, which is a regular Croatian female name and can be associated with the city.

Project goals:

The system is planned to provide the service as the follows: when a visitor comes to the system, Dubravka recognizes the visitor as a Western person, Japanese, or Croatian from a combination of the speech recognizer's result and the nonverbal behaviors of the visitor. An example of such obvious cues is bowing, which Japanese people use for greeting. The agent then adapts itself to the Japanese mode, that is, it speaks in Japanese and behaves in Japanese ways to provide the visitors with tour information. At the same time, visitors can interact with the agent not only by speaking in their natural language but also by nonverbal gestures and posture behaviors such as pointing to an object in the background image or by raising their hand to indicate that they want to ask a question.

Task distribution:

From the nature of ECA development with GECA, the system is composed with a number of standalone components which are not tightly bound to each other. Therefore, each component can be assigned to one team member according to his(her) interests and ability without heavy dependency with the other members.

- Member A
  - Project management
  - GECA platform improvements
  - English/Japanese speech recognition
  - Japanese version of sightseeing information scenario (translation)
- Member B
  - Character animation player improvements like scene transitions
  - Croatian speech input/output
- Member C
  - Sensory devices and the recognition of nonverbal inputs
- Member D
  - English/Croatian version of sightseeing information scenario (original)
- Member E
  - Tour guide data collection in Japan and Croatian
  - Literature investigation on cultural differences of gestures
  - Nonverbal animation creation of the Dubravka character

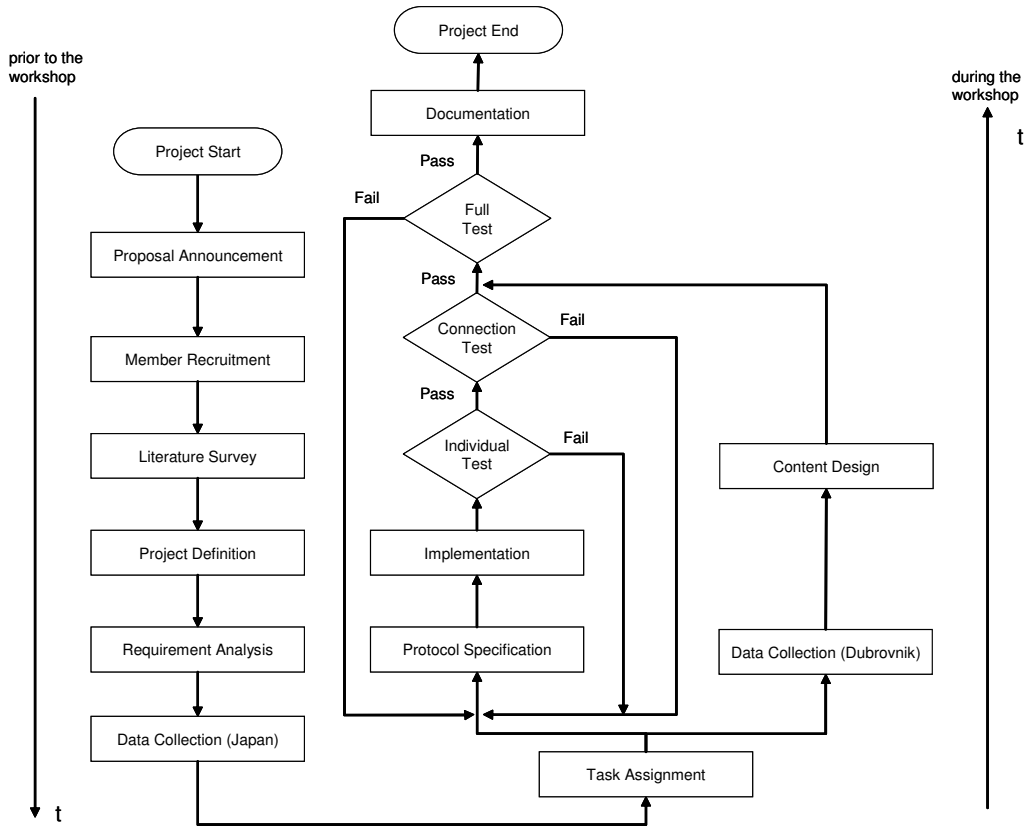

Figure 4.1: The workflow before and during the eNTERFACE'06 workshop

Project schedule:

The workflow of the development of this project is shown in Figure 4.1. The process is planned before the workshop until the end of the workshop. Since the objective and requirement is not very obvious in the beginning of the project, each component are tested with the components which it is connected and the interface protocol is refined if there is any flaw.

The development of Dubravka does not obey but is similar to agile software development method (Beck et al., 2001). All of the project member meet together at the same place for one month. The project time is very limited so the timebox is scheduled as every week as the following schedule. The members met each other everyday and can make face-to-face communication to immediately correct their responsible tasks if necessary. A meeting of the whole group is held in the beginning of each week. The result of last week is reviewed and

the scheduled tasks of that week is checked and can be modified/refined in the meetings.

- First week:
  - Design of each components and discussing the interfacing protocols
  - Implementation of the English speech recognizing and synthesizing components
  - Gather the verbal and nonverbal behaviors (e.g. head nods, facial expressions, and hand gestures) of human tour guides. For example, in the organized social events of eNTERFACE'06, record and investigate how people greets, catch attention before asking a question, understood an explanation, release of utterance turn, and say goodbye.
- Second week:
  - Implement the software components
  - Build the necessary animation/action database
- Third week:
  - Implement the software components
  - Connect the software modules with our GECA framework and test whether the application work properly
- Fourth week:
  - Debug and improve the system
  - Prepare the final demonstration
  - Jointly write the final report of this project

Project results:

Because of the nature of the eNTERFACE workshop, there were two general difficulties for each team to achieve its goal.

- The four-week period of the workshop was relatively short to realize significant achievements or start a new project.
- There were some team members who were not directly engaged in this joint research project or not familiar with the fields which the project involved.

Reducing the hurdles for the team members and minimizing the effort of developing new programs were thus essential issues for producing as many results as possible in the limited four-week workshop period. The project benefited from the GECA framework but not all of the scheduled objectives could be completed before the end of the workshop. All of the individual components are completed, but they were not integrated during the workshop. Although not all of the ambitious objectives of this project could be achieved during the period of eNTERFACE'06, we continued developing it after the workshop.

4.2 The Investigation on Cultural Issues

Culture is relevant to many aspects of human-human communications. These effects should be also reflected throughout the design of culture-sensitive ECA systems: how the agent interprets its perceptions, how the agent thinks, and how the agent behaves. From the point of view of communication interfaces, the language spoken by the agent directly determines how the user perceives it and is an obvious factor that distinguishes different cultures.

Cultural differences are also displayed in people's nonverbal behaviors. The same gestures may represent different meanings in different cultures and the same meaning may be represented by different gestures. Sometimes the differences are coded culturally, for example, beckoning gestures are displayed in exactly opposite directions by British and Japanese people. The finger gestures representing numbers provide another example; Japanese people use two hands and overlap one of them with the other one while Chinese people use only five fingers of one hand to present numbers from one to eight, even though these two cultures are similar in many aspects. Misuse of these culturally coded emblem gestures may cause misunderstandings and problems in communication.

Handling cultural issues is very relevant to emotion control and the deliberations of the agent (Rosis et al., 2004). However, in a four-week project, it was not possible to explore

these issues in depth. In the case of the Dubravka agent, we were only able to handle the surface of cultural issues, i.e., the perceptions and behaviors of Dubravka including the language that the agent spoke and listened to, and the usage of different culturally coded emblem gestures.

A significant feature that has not yet been achieved is the automatic recognition of the culture class to which a user belongs from her (his) nonverbal behaviors. We realized that it is difficult to find the differences in nonverbal behaviors between users coming from different cultural backgrounds since the beginning of the interaction with the agent. This is an extremely difficult task even for humans and more research is required. Instead of that, the current system is switchable to different culture modes by asking the user to select a cultural mode with a question in English at the beginning of interaction.

Since our target is a tour guide agent who serves visitors from Japan, Croatia, or somewhere in the Western culture area, the first task was to gather culture-specific behaviors in the tour-guiding context, particularly the culturally coded emblem gestures. The material we used was mainly obtained by taking video data of Japanese tour guides at several famous sightseeing spots in Kyoto and European tour guides in Dubrovnik (Figure 4.2). Appropriate nonverbal behaviors of the agent were chosen from observation of the collected video corpus and the ones introduced in (Hamiru.aqui, 2004).

While modeling the gesture styles for the character, we aimed to emphasize the diversity of the three cultures. For example, we introduced the “cross hands in front of the chest” gesture in the Japanese mode. This gesture is usually performed with additional head shaking to express negation. It seems to be rather unique and normally draws the attention of Western people who first come to Japan (Figure 4.3 left). Another example is the “prohibition” gesture (Figure 4.3 right). In Japan, it is expressed by waving with a hand while the arm is extended. Sometimes shaking the head sideways is also added. When asking to wait, Japanese people usually show the palm of one hand to another person. At times, both hands could be used. Some confusing gestures can make people misunderstand because of different interpretations in different cultures. For example, the beckoning gestures that mean “go away” and “come here” are performed in opposite directions in Western countries and Japan. In Dubravka’s Croatian and general Western modes, she gestures “come here” by waving upwards and backwards with one hand and the back of the hand facing downward.

Figure 4.2: One scene collected in the tour guide video data, one of Croatian workshop participants is introducing Dubrovnik city to the others and is performing a beat gesture

However, this gesture may be interpreted as “go away” in Japan. Therefore, in her Japanese mode, this gesture is performed with the back of the hand facing upward.

Unlike the Japanese gestures, which are often significantly different from the Western ones, we could not find obvious differences among the Western tour guides, even if they came from different countries, in our observation of the video corpus. Table 4.1 shows some examples of gestures modeled in the Dubravka agent system.

4.3 Building the Dubravka Virtual Tour Guide Agent

The functionalities of the Dubravka agent system are divided into standalone GECA components so that each one of them only supports relatively simple functions and they are loosely coupled with each other. The components then jointly generate the behaviors of the tour guide agent as a single integral system. By this approach, the number of necessary newly developed programs can be decreased and legacy components can be reused without significant modifications.

In the Dubravka agent, some components like the animation player or the sensor devices can be the same in the three different cultures, and some parts like speech I/O or culturally-coded emblem gesture animations are similar but different in the three cultures. The system

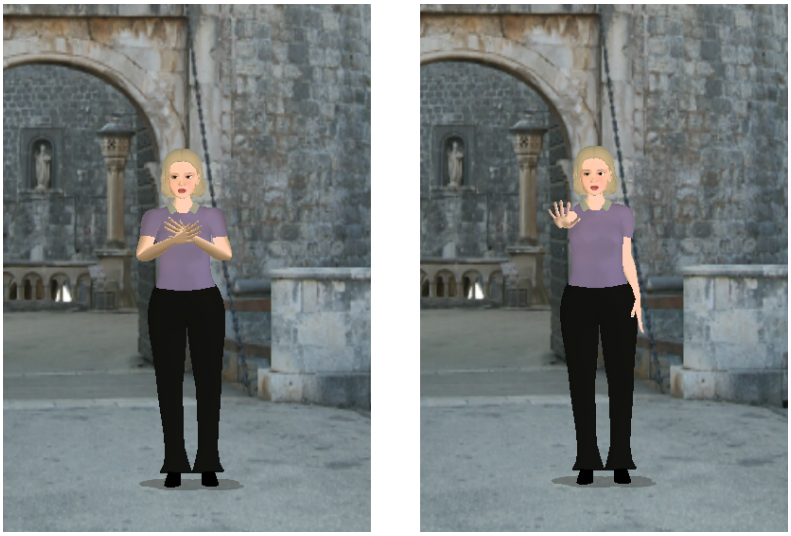

Figure 4.3: Tour guide agent Dubravka showing Japanese *negation* and *prohibition* gestures

can benefit from being composed of culture-dependent components which are dynamically switched to the currently appropriate ones according to the cultural mode while culture-independent ones are shared and are always running across different culture modes.

The system was built by reusing as many available components as possible to reduce the efforts required to develop new components. The following is an inventory of the software components and the contents used in the Dubrovnik tour guide application.

The components which can be reused by another ECA system:

Scenario component. This component is an implementation of the GSML script interpreter. The available interactions with the human user in three language modes are defined in a single script.

Japanese spontaneous gesture generator component. This component is a wrapper of the CAST (Y. Nakano et al., 2004) engine which generates the type and timing information of spontaneous gestures from a Japanese utterance input string. This component has been implemented.

Table 4.1: Some examples of the differences of the gestures displayed in each culture mode

Action	Culture dependency	Croatian	Japanese	Western
Bow	In this gesture, we present three types of bowing: shallow bow, using only head; deeper bow (Japanese style) shows respect to the listener	√	√	√
Invite	Croatian and general Western gesture presents waving upwards and backwards with one hand and the back of the hand facing downward. However, this gesture may be misunderstood as "go away" in Japan. In Japanese mode, this gesture is performed with the opposite orientation of the back of the hand	√	√	√
Cross	This is a Japanese emblem gesture, meaning that something is not allowed. The hands are crossed in front of the lower part of the chest		√	
Extend	This gesture means right arm extended with the palm open and oriented upwards. In the Japanese culture it means "wait please"		√	
Wave	This gesture presents oscillating right hand waving. Used in combination with the "extend" action as part of the Japanese gesture meaning "No"	√		√
Banzai	Throwing both arms up expresses good fortune or happiness		√	

Character animation renderer component. This component is a wrapped character animation player that is implemented with visage|SDK. It accepts driving event messages from the animation category and speech synthesizer component and performs the specified character animation. Because the character animations need to be synchronized with voice with a precision of milliseconds, Text-To-Speech (TTS) engines must be tightly bound to the player. In the current implementation, English and Japanese words that the agent speaks are generated by Microsoft SAPI compatible Pentax VoiceText (Hoya Corp., 2008) TTS engines.

English and Japanese speech recognition components. These components are wrapped recognition engines that recognize Japanese or English spoken by the visitors by matching predefined grammar rules. Because of the lack of a good enough speech recognizer for Croatian, it is recognized by an English speech recognizer with grammar rules, which will be explained later in this section.

Sensor data acquisition components. The nonverbal behaviors of the users are recognized

by using the data from data gloves, motion capture, head tracker, and acceleration sensor. In eNTERFACE'08, two new components were introduced. One detects whether there are user movements by using OpenCV (Intel Corp., 2006) and standard image difference techniques are also implemented. The other uses a commercial product, Omron's OkaoVision (Omron Corp., 2008). It is a library that provides accurate face detection and extra functions like face orientation, gaze direction, the positions and openness of eyes and mouth, gender detection, age identification, and face identification from a single image. It has the inherent limitation that when the users turn their heads to the left or right then their faces cannot be detected. These components acquire raw data from the sensor devices, interpret them, represent those events as text strings and send the results to other components for further processing. The configuration of these hardware devices is shown in Figure 4.4.

Input interpreter component. This component was introduced to combine the raw data from several sensor components to generate the event messages that can be processed by the scenario component. The task of this component is sensor-dependent but application-independent. In the current system, it combines the raw data from the data glove and from the motion capture to generate user pointing positions and combines data from a motion detecting component and the OkaoVision component to detect the exact number of users present.

The contents need to be specifically created for the Dubravka agent:

GSML scenario script. A GSML script describing the anticipated interactions between the agent and the user in the tour guide context must be created specifically for the application. Currently, the script includes a scenario in three languages (English, Japanese, and Croatian) and possible human-agent interactions in five different scenes: the entrance gate of the Dubrovnik old town, a fountain, a monastery, and two other scenes in Dubrovnik's main street.

Background images and the positions of the agent. The background images and the coordinates where the agent should stand and can walk to need to be prepared for each scene. The appropriate positions, size, and orientation are computed with ARToolkit

(Kato, n.d.).

Croatian voice tracks. Because of the lack of a Croatian TTS, the agent's Croatian speech is recorded from a native speaker's voice.

Speech recognition grammar. Speech recognition in the current system is keyword-based and the grammar for recognizing those keywords needs to be prepared.

Additional character animations. Additional character animations which are not available in the animation catalog need to be prepared.

The components which are limited to use in this tour guide agent:

None. Although some of the system components were developed in the workshop, they can be used in other applications because of their simple and well-divided functionalities.

The data flow among the components is shown in Figure 4.5. The cost of building a tour guide agent that is adaptive to three cultures can be kept low. In the current system, all of the components are culture-independent ones. The scenarios of the three cultures are represented in the same script, but each conversational state in GSML is labeled with a language attribute so appropriate TTS and nonverbal behaviors will picked automatically by the scenario executor. The only exceptions are the speech recognition component; one recognition component is required for each different language and only the results that match the currently valid language will be processed. The following subsections introduce the tasks done for incorporating the three cultures into the tour guide agent.

4.3.1 Nonverbal User Inputs

Because advanced gesture recognition techniques have not been introduced, in the nonverbal input recognition part the system is not recognizing culture-specific nonverbal behaviors from the user but only the following general ones at this moment:

- pointing to the interesting objects shown on the display
- showing the wish to ask a question

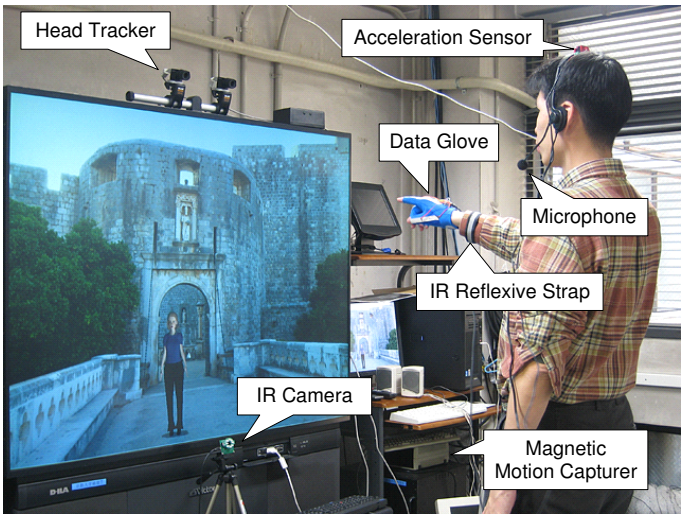

Figure 4.4: The hardware configuration of the multi-modal Dubrovnik tour guide agent

- interrupting the agent's utterance
- shaking the head and nodding to express negative and positive answers

These behaviors are recognized by combining the data from the sensor devices. For example, a pointing gesture is recognized by a pointing shape from the data glove and the pointed positions on the display from the coordinate values of motion capture. The movement detection component and face detection component are used to generate the exact number of available users. Because each type of raw data is not meaningful to the central scenario component, the input interpreter component is responsible for generating the combined information, the position where the user is pointing, for processing by the scenario component.

4.3.2 Character Animations

Some of the gesture animations are created by programming routines that generate joint parameters during run-time. Since we did not have a tool to translate real human gestures into the set of animation parameters in the CG character player, we had to create animations manually. This was a rather time-consuming approach; it took about 5 to 30 experiments to adjust the parameters for one action, depending on the complexity of the action. Although

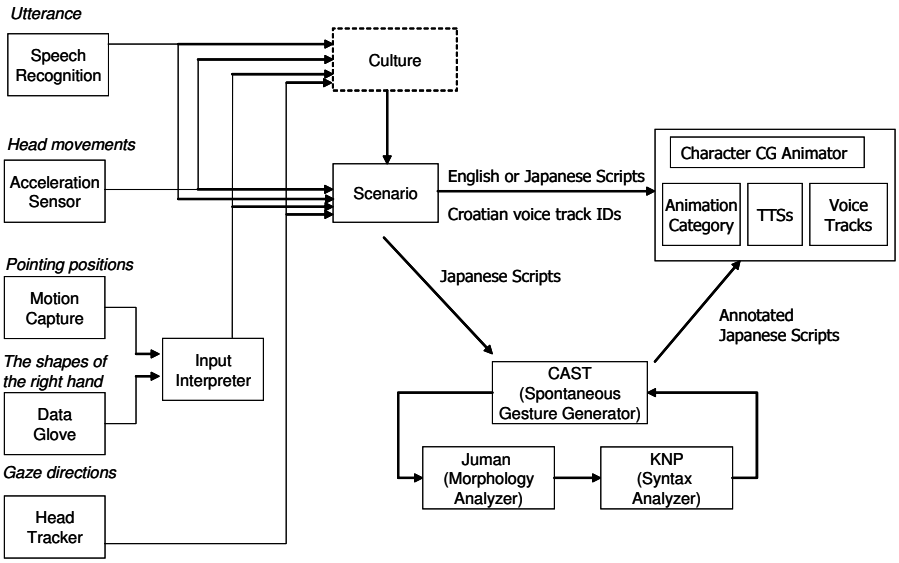

Figure 4.5: The data flow and component configuration of the multi-modal tour guide agent. The programs, CAST, Juman and KNP communicates with each other in their original protocols. The dashed box labeled "Culture" is not implemented yet

routine-generated gesture animations have the disadvantage of a relatively unnatural look, they have the advantage that the duration of the animation can be determined at run-time. Locomotion animations have to be implemented by programming. On the other hand, some gestures are modeled in the 3D CG modeling software Autodesk 3ds Max (Autodesk, 2009); they look more natural but their duration is fixed. Currently, we have 20 routine-generated gesture animations and 27 animation sequences that are modeled in 3ds Max with fixed lengths. Some of these gestures are shown in Table 4.1. Since most of the Croatian gestures are also used in many European cultures and in general Western cultures, we did not distinguish them in the current implementation.

4.3.3 Croatian Speech Input/output

Although Croatian is spoken by around five million people, the commercial speech and language communities have not yet produced general purpose recognizers, synthesizers, and translation engines for the Croatian language. This section describes the alternative solutions adopted in the development of Dubravka's Croatian mode.

Croatian speech input

In the field of Croatian speech recognition, some research studies have been done, but none of them have produced general purpose recognizers. (Ipšič et al., 2003) and (Peic, 2003) developed a bilingual database of Slovenian and Croatian weather forecasts. Their recognition results for the two languages are very similar and in the future, they plan to perform bilingual speech recognition system simulation. Nevertheless, a Croatian speech recognition engine is still unavailable to the research community or to industry. Therefore, we decided to configure an English speech recognition software component to recognize Croatian speech by tailoring the recognition grammar. Within the system, classification of the user's utterance is done with limited vocabularies of specific keywords spoken by the user that trigger the scenario component. The pronunciation of Croatian keywords in scenarios is approximated by using the English alphabet. Since some Croatian words in the scenario were impossible to represent in the English alphabet, we had to choose other words instead. If the grammar contained similar words, those words sometimes confused the recognizer, so we were careful to choose words that are not too similar. For example, the pronunciation of the Croatian word "da" (in English: yes) is approximated in the English alphabet as "ddhaa". Although the speech recognizer works well with the recognition of the word "da" in Croatian, it is often confused by words that contain the syllable "da", like "slobodan" (free). We therefore could not choose short words like "da" or "dan" (day) that can appear in longer words, and thus the Croatian scenario is slightly different from the English and Japanese ones. In the end, the following two principles were followed in choosing words to compose the Croatian scenario. The keywords approximated with the English alphabet are not very short and do not contain the syllables of other keywords. Table 4.2 shows Croatian words used for recognition and the corresponding pronunciations of those words represented in the English alphabet. Because there are only five scenes in the current system, transitions between the scenes and between the states in each scene do not require many keywords from speech input. In the English and Japanese scenarios, we used eight words for transitions and seven of them in Croatian.

Table 4.2: Croatian words and their approximated English alphabets used in speech recognition

No.	Croatian word	Meaning in English	English alphabets
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

Croatian speech output

Since there is still no available Croatian TTS with satisfactory quality, Croatian speech output can only be implemented with a recorded human voice. After the Croatian scenario was composed, a native Croatian speaker's voice was recorded to prepare all the utterances that are supposed to be spoken by Dubravka. The recorded voice tracks are paired with lip animations that are generated automatically by (Zoric & Pandzic, 2005). The speech signal is derived from a type of spectral representation of the audio clip and is classified into viseme classes by using neural networks. The visemes are then mapped to MPEG-4 facial animation parameters and are saved as MPEG-4 FBA tracks when the Croatian speech utterances were being recorded. They are then played by the player with synchronized timings at run-time.

4.4 Potential Extensions

The Dubravka agent built in the eNTERFACE'06 workshop was relatively simple and only addressed the surface issues of multi-culture competent ECA. In this section, we would like to discuss possible extensions to it.

4.4.1 Training or Pedagogical Purposes

Another possible extension is use for training or pedagogical purposes. Figure 3.9 shows another system that we developed for experiencing the differences in gestures between different cultures. There is an avatar that replays the user's hand gestures, such as beckoning, while ten computer-controlled agents react to those gestures differently pretending that they are Japanese or British. The user's actions are captured by a magnetic motion capturing device and interpreted to low-level joint angles to drive the avatar character in real-time. The computer controlled agents are driven by individual reflexive controlling components and a common BAP catalog component. They are driven by low-level MPEG-4 BAPs in real-time, too. We would like to incorporate this extension into the Dubrovnik tour guide system in the future.

4.4.2 Culture Module

One of the benefits from the modular and distributed design of GECA is that extending the current system to incorporate another culture at the same detail level is straightforward. The developers only need to prepare the speech recognition and TTS engine for that language, additional character animations if required, and the scenario script. In addition, the dashed "Culture" box depicted in Figure 4.5 is a potential extension of the current system with a culture module.

In addition to emblem gestures, as suggested in the CUBE-G project (Rehm, Andre, et al., 2007; Rehm, Gruneberg, et al., 2008; Rehm, Nakano, et al., 2008), the cultural class to which the user belongs to can potentially be inferred from the characteristics of the user's non-verbal behavior. The classification criteria can be collected from empirical and statistical results. For example, how frequently the user performs gestures, the strength of the gestures, the distance from the agent chosen by the user, and so on could be informative.

The culture module can then be built to accept the sensor data from the non-verbal input modules, analyze their characteristics, and then classify where the user come from according to a Bayesian network (Rehm, Bee, et al., 2007). The results from speech recognizers certainly provide clear evidence of culture. The classification result from the culture component can then be sent to the scenario or deliberation component to affect the characteristics

of the agent's behaviors in a parameterized way, for example, done faster or with a larger spatial extent. (Solomon et al., 2008) have proposed a language for describing ethnographic data in a pluggable design that could be a candidate of the internal representation of the culture component.

4.5 Extend Dubravka to Interact with Two Users Simultaneously

It could be a complex but interesting challenge to combine the multi-user and multi-culture tasks. What should the agent do if the users do not belong to the same culture class? In the eNTERFACE'08 workshop (Cerekovic et al., 2008), we investigated multi-party interaction issues and improved Dubravka to be able to interact with at most two users. The hardware configuration of Dubravka agent is shown in Figure 4.6. and the system architecture is shown in Figure 4.7. Each of the user wears a headset for speech recognition by using Loquendo ASR (Loquendo Corp., 2008), two cameras are set up for detecting the users' activities by using a face detection library, Omron's OkaoVision (Omron Corp., 2008) and skin color detection with OpenCV (Intel Corp., 2006), the results of these two components are then combined in the input understanding component. The possible dialogs of this system is driven by a predefined GSML script where the interaction rules with the visitors are described. The script is interpreted by the *scenario* component, and the agent's actions are triggered according to the results from the input understanding component. The scenario component also drives the character animator to play animations. During the interactions with the users, Dubravka always keeps the initiative, but the users can interrupt her and ask questions about current topic. We designed the topic related questions and by using keywords "where," "when," "how" what are defined in the speech recognition engine's grammar rules. Dubravka also asks the users simple "yes/no" answer-based questions. In the eNTERFACE'08 Dubravka agent system, the following issues were investigated.

Appearance of users. In the ECA systems which multiple users can be present, it is important to detect and locate the user(s) in order to achieve natural conversations. The system uses image processing techniques to recognize motions, facial positions and orientations.

It can recognize dynamically changing user numbers with maximum number two and their positions.

Channel management. The system needs to combine users' speech with nonverbal behaviors to resolve the ambiguities among multiple modalities. In current prototype, the nonverbal behaviors taken into account are face orientation and gaze directions. By combining these modalities, the system is able to distinguish one of the following situations, decreased level of attention, making requests to the system, leaving the system or speech collisions.

Distinguishing conversational roles. While Dubravka is talking to both of the users, she gazes at both users with the same frequency. In the cases Dubravka is talking to one of the users, she gazes more frequently at the addressee and gazes less frequently at the other user whom are treated as the overhearer. The rule to determine which user is the addressee is as the followings, when Dubravka is presenting a new information on her own, she treats both of the users as the addressees and treats one of the users as the only one addressee when she is answering a question asked by that user. Because each user wears a dedicated headset so the system can identify who is speaking and who asked a question. Loquendo's ASR (automatic speech recognition) (Loquendo Corp., 2008) engine is used in the speech recognition component because its high tolerance to noise so that the speech of a user can be correctly recognized even there is another person speaking besides him or her.

Handling the conversation between the users. The situation when the users started to talk with each other is detected and handled in the Dubravka agent system. It is detected by the face orientations and the timing of speech, i.e. when the two users are speaking and face to each other. Since Dubravka has only limited conversation abilities, she can not join the conversations occurred between the two users directly. Instead of that, she detects the user conversation and tries to get their attention back by actions like proposing the change of topic.

During the eNTERFACE'08 workshop, we also noticed that the agent shown on a 2D screen can not effectively convey its attention like gaze or pointing gestures to the addressee. This is known as "Mona Lisa Effect" and is intensively investigated in (Morikawa & Maesako, 1998) but is not explicitly addressed in the ECA community yet. For example, both of the users will feel being looked at if the agent is looking straight forward. The agent's

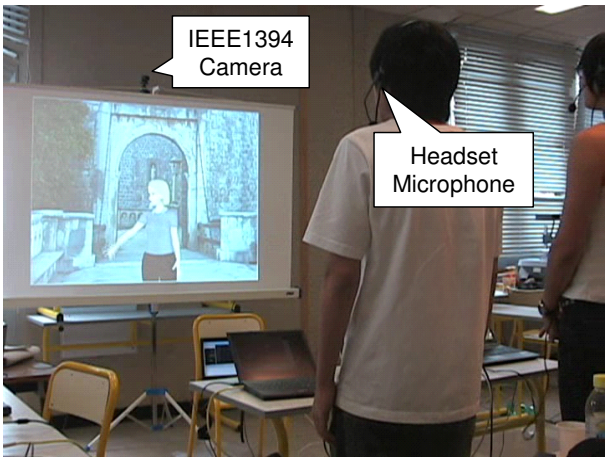

Figure 4.6: Dubravka interacting with two users in the eNTERFACE'08 workshop

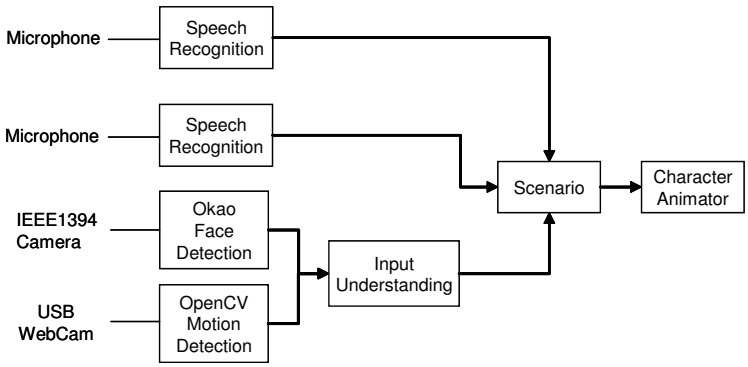

Figure 4.7: The system architecture diagram of the eNTERFACE'08 tour guide agent

attention can only be perceived (actually inferred) with extreme conditions, i.e. the case when there are only two users, large size agent shown on the screen, the users standing close distance to the screen, and large distance between the two users (Figure 4.8). Obviously, according to the system settings, this issue can have significant influences on the users' perception.

4.6 Discussion and Conclusion

ECAs are very useful tools for representing cultural differences in training and edutainment applications. In this chapter, we have presented preliminary results from the development of

Figure 4.8: The extreme condition of an agent shown on a 2D screen that allows the users to feel the gaze direction matching the agent's conveying attention. In other cases, the attention of the agent will be ambiguous, both of the users may feel that the agent is looking at themselves or is looking at the other user

our culture adaptive tour guide agent system that is implemented in a modular way with the GECA framework to minimize the development cost. It can switch its behaviors and speech language to three culture modes: general Western, Japanese, or Croatian. Although both the tour guide agent and GECA itself are still in relatively early stages of development, this very loosely coupled and modular framework can have three possible benefits in handling cultural issues.

- Culture researchers who are not familiar with technical issues can introduce ECA technology more easily because they need only concentrate on culture-dependent issues and implement them as a separate component. The component can then be integrated into a culture-independent skeleton ECA for quick enculturation.
- Collaborative studies with research teams from several countries can separately implement their own culture module more easily.
- Research efforts done in the analysis by synthesis style can be refined incrementally more easily.

This study focuses on the rapid building of ECAs and only features the surface traits of culture, that is, languages, emblem gestures, and probably culture-dependent characteristics

of gestures. A more thorough study based on theories of inter-culture communication is necessary in the future. For example, we have noticed that in the case of an interface ECA serving Japanese and Western users, the high-context/low-context differences proposed in (Hall, 1992) should cause obvious differences in the behaviors of real humans. Nevertheless, our system models the agent behaviors in a one-to-one mapping sense; the agent always do something in Japanese mode or its counterpart in Croatian mode, even though real Japanese people and Croatian people might make totally different decisions in the same situation.

By using scripts to describe human-agent interactions, the range of possible interactions will be relatively limited and the quality of the whole system heavily depends on the knowledge and skill of the agent designers. At this moment, we are only showing the feasibility of our modular approach. Obviously, this is not yet a sound solution, but we would like to further develop the deliberative part of the agent with culture modules that affect its outputs with culture-specific differences, and to explore the high level aspects of cultural issues like the use of silence during dialogue, intonation, the choice of words, and so on in the future.

Finally, in section 4.5, the following two main insights in multiple-user setting are mentioned. First, the participants may interact with each other. Second, there should be some way to let the users to distinguish the 2D agent's attention. These insights are further utilized in next chapter.

Chapter 5

Quizmaster Agents for Real-world Exhibitions

This chapter describes our developments on ECAs as quizmasters by using GECA framework. They are started from the collaborative project with the National Food Research Institute (NFRI) of Japanese government. NFRI is executing research programs that contribute to secure supply of safe food, and technical innovation in agriculture and food industries. The research programs include clarification and utilization of functional properties of foods, development of innovative technologies for food distribution and processing, development of techniques to ensure food safety, and development of technologies for biomass conversion. At the same time, this institute also bears the responsibility to be the source of dispatching food information and arouse the public's awareness on food safety. To achieve these goals, it holds open lab fairs and participates in exhibitions related to the Ministry of Agriculture, Forestry and Fisheries (MAFF) every year.

In these events, the direct concern of the staff of this institute is how to attract more visitors, and how to improve the efficiency in knowledge transfer of their research results and general tutorial materials to general public audiences. We jointly sought for the technology to facilitates this task. After a series of discussions, we conducted to build a quiz game kiosk with an ECA behaves as the quizmaster (hereafter quiz agent) based on the following hypothesis: comparing to static exhibits, an interactive exhibition which the visitors can participate in should be more attractive, enjoyable, impressive and has higher chance to

stimulate the visitors' interest and consciousness in food science. Quiz was then chosen as the target game application because it is very popular as learning materials or TV programs. Especially in Japan, people are familiar with quiz media and no further explanations are required. While at the same time, a quiz agent who issues quizzes and explains the correct answers should make the interface more intuitive for general public visitors and the quiz contents more comprehensive.

Visitors in groups is an inevitable situation that the ECAs placed in public exhibitions have to face but is seldom addressed in previous systems. Our exhibited prototype of NFRI quiz agent also relies on the participants' direct operation from the touch panel and is not aware of the multiple participants. To improve the interaction experience of the participants with the quiz agent and to improve its life-likeness, we are motivated to make the quiz agent attentive to the status of multiple participants' activities during the interactions.

This chapter first introduces the NFRI exhibitions of a simple quiz agent (section 5.1). Then the preliminary investigation on the observed interactions occurred among the visitors and the agent in, and the proposal of two approaches in realizing multi-user attentive quiz agents (section 5.2). The first one utters at appropriate timing at appropriate addressee based on a policy that is triggered by the estimations on the activity of each participant and the whole group (section 5.3). The second one introduces a transition state model of the agent's internal attitude toward the participants' status. This attitude drives the nonverbal animations and utterance of the agent when it stands by (section 5.4). To evaluate these two prototype systems, we introduced the use of GNAT (Go/No-go Task) test as well as regular questionnaires and video analysis (section 5.5).

5.1 NFRI Quiz Agent Exhibitions

The first NFRI quiz agent prototype is composed as Figure 5.1, the quiz agent stands at the right hand side of the window while the subtext of the quiz question and the answer candidates are shown at the left hand side of the screen. The quizzes are selected randomly from a quiz database containing 43 (2007) and 71 (2008) quizzes in total. The quiz kiosk is set up as Figure 5.2, the application is projected to a large screen, and a touch panel was chosen here as the user interface for the convenience in public exhibitions. For visitor

perception in life-likeliness of the agent, it sometimes walks in the scene, does pointing gestures on the answer candidates and shows facial expressions. In addition to these, the BGM (background music) changes in reflecting the status of the game progress. The quiz game progresses as the following phases:

1. The quiz agent greets the game participants and explains the game rules.
2. The quiz agent issues the quiz question.
3. The quiz agent stands by until the participants press a graphical button on the touch panel to answer.
4. The quiz agent gives the participants hints if they press the hint button shown on the touch panel.
5. The quiz agent announces the correct answer after the participants pressed one of the answer buttons on the touch panel.
6. The quiz agent gives the participants a comment about the answer or the difficulty of this quiz question.
7. The quiz agent ends the whole session after 10 quizzes by giving the participants a summary about their performance.

This prototype is built by combining standard GECA components, GSML executor, animator and the GECA server with two additional components, a touch panel component and an emotion component. The component configuration of the quiz agent is shown in Figure 5.3. The change of BGM is controlled by the emotion component. The methodology is inspired from MAX agent's emotion simulation mechanism (Becker et al., 2004) that is based on the PAD (pleasure-arousal-dominance) model (Mehrabian, 1996). The emotion module gets positive stimulation on emotion and mood axes when the participants pressed the touch panel to answer the quiz, they get even higher values if the answer is correct but perceive negative stimulation when the answer is wrong. The value on boredom axis grows when there is no input from the visitor for a while. The emotion component continuously changes its internal state to play 14 background melodies depending on current state like

Figure 5.1: A screen capture of the NFRI quiz agent

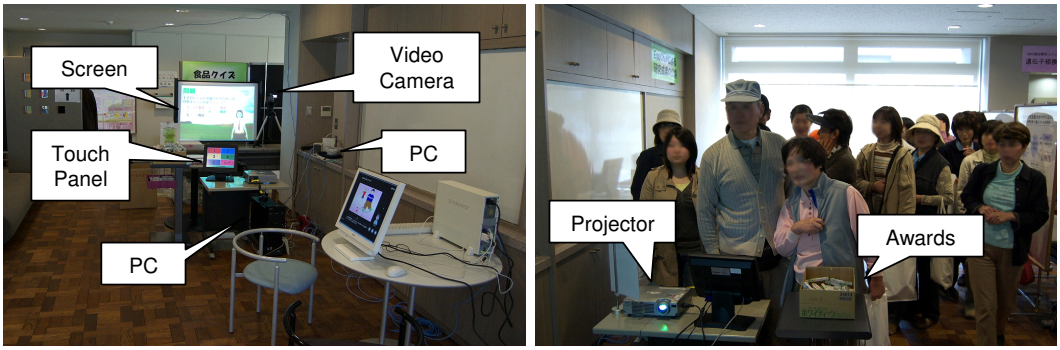

Figure 5.2: The configuration during the first exhibition of NFRI quiz agent

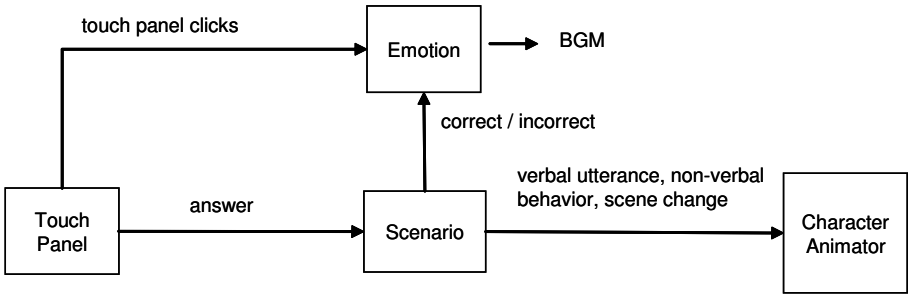

Figure 5.3: The architecture of the first generation quiz agent without user awareness

Table 5.1: The summary of the NFRI exhibitions where the quiz agent is displayed. Here, GN, PN and GS denote the number of groups who participated the quiz game, the total number of participants, and the average size of each group, respectively

Exhibition	Date	GN	PN	GS	Participants
Open Lab.	Apr. '07	87	307	3.52	students, house keeping wives, elderly people, couples
Agriculture expo.	Nov. '07	55	109	1.98	agriculture experts, house keeping wives
Open Lab.	Apr. '08	70	237	3.38	students, house keeping wives, elderly people, couples
Open Lab. for summer vacation	Jul. '08	78	207	2.65	parents and children

angry, bored, concentrated, friendly, etc. The facial expressions, however, are not changed dynamically according to the internal emotion state of the agent but are shown statically as they are defined in the GSML script according to the status of the game.

This quiz kiosk was shown in four NFRI exhibitions held from April 2007 to July 2008. Table 5.1 summarizes these events. The typical visitors of these events were the people who live in the neighborhood or teenage students come from nearby high schools. The exhibitions were six-hour long for one day each time. Almost during the whole day, there were dozens of visitors waiting for playing the game every time. Therefore, we considered that the basic idea was very successful in attracting the visitors.

To get an insight of the exhibitions, we added a questionnaire session to investigate the life-likeness and multi-user capabilities of the quiz agent in the two events of 2008. The same as the quiz game, the questionnaires are answered by the users in groups as an option

Table 5.2: The five-scale questionnaire results gathered in the two open lab events during 2008. The data (50 groups) collected in the exhibition for public consumer are listed in the upper row, and the ones (53 groups) of the open lab event for parents and children are shown in the lower row. The numbers mean the number of groups and the ones inside the parenthesizes mean percentages

	5	4	3	2	1
Our decisions were influenced by the character.	21(42.0) 13(24.5)	11(22.0) 21(39.6)	10(20.0) 5(9.4)	3(6.0) 8(15.1)	5(10.0) 6(11.3)
The character was human-like.	13(26.0) 8(15.1)	6(12.0) 28(52.8)	9(18.0) 5(9.4)	8(16.0) 11(20.8)	14(28.0) 1(1.9)
The character was aware of us one by one.	9(18.0) 6(11.3)	8(16.0) 20(37.7)	8(16.0) 12(22.6)	8(16.0) 10(18.9)	17(34.0) 5(9.4)
We enjoyed the game.	31(62.0) 34(64.2)	15(30.0) 17(32.1)	0(0.0) 1(1.9)	3(6.0) 1(1.9)	1(2.0) 0(0.0)

after the game itself. Finally, we gathered 50 and 53 results respectively. The other five-scale questions are listed in Table 5.2.

The questionnaire results implied that the methodology to adopt a quiz agent in exhibitions was successful in attracting visitors, entertaining them, and as a result that they get interested in the exhibits can be expected. During the whole game session, the developers of the system was beside the participants, explained how to use it and how to fill the questionnaires if necessary. Even though in this situation, the participants may tend to answer the questionnaires in favor of the developers (agents), there was still a considerable number of participants had negative impression on the aspects of human-likeness and user-awareness of the agent. This was particularly obvious in the exhibition for public consumers. The quiz agent can be improved in human-likeness and user awareness, especially in responding to multiple participants at the same time.

5.2 The Two Approaches to Realize the Multi-user Attentive Quiz Agent

The first prototype of NFRI quiz agent is obviously unable to respond to the status of the participants. The is due to the lacking of sensing mechanism, the agent is not aware of the game participants no matter they are in groups or come individually. To further improve the quiz agent system, an official evaluation of it is required but could not be easily done due to the inherent of general public exhibitions: Many participants are teenage students who do not have legal rights on their own behaviors. There are always dozens of visitors queuing to participate in the quiz game. The general public visitors do not have the knowledge about ECAs or researches. It was impractical to explain the experiment objectives to the participants and to obtain the authorizations to collect and analyze the data formally and systematically.

Instead of that, from the observations on the participant-agent and the participant-participant interactions during the exhibitions, we have the following findings:

- Most of the visitors participated the game in groups and answer the quizzes as a collaborative task of the group members via discussions .
- The activity of participants' interactions changes dynamically, i.e. sometimes participants discuss very actively, but sometimes they think about the answer separately.
- There is usually one or more participants who leads the conversations involving the discussions and negotiations on the final answer of the quiz. This person(s) may change among different quizzes.
- The participants guffaw or exclaim when the announced answer is surprising or when the agent says or does something silly, e.g. a strange and unnatural pronunciation from text-to-speech engine or an awkward gesture.

Because of the limitations comes from natural language understanding with contemporary technology, it is difficult for the agent to actively join the conversations of the participants. Nevertheless, from the findings above, by utilizing the dynamically changing activity

of the participants' conversations, the participants' feeling of the agent's attentiveness seems to be achievable in the quizmaster task. In order to realize an attentive quiz agent who is aware of and adapts its behaviors according to the status of multiple participants, two aspects of attentiveness that are complementary to each other can be considered:

As the effects on the agent's intentional behaviors toward the world external to it. These effects include when the agent should utter, what the agent should utter, and who is the addressee of the utterances of the agent.

As the effects on the agent's own attitude but expressed as observable behaviors. These effects include what the gestures and behaviors expressed by the agent are.

At the same time, we do not have very concrete ideas about how the agent should behave to make the participants perceive that the agent is attentive from its behaviors. In order to explore the effects of these two aspects more thoroughly without interfering each other, two variations of improved quiz agent (attentive quiz agent A and B) are then developed with corresponding hypothesized strategies and are evaluated, respectively. After a detailed investigation of the affective factors, we plan to integrate them into an individual attentive quiz agent that can be practically deployed in exhibition events. Therefore, the use of sensor devices is kept minimum as a prerequisite and only video/audio information are used to estimate the users' status.

5.3 Quiz Agent Who Utters Attentively to Multiple Participants (Agent A)

For attentive quiz agent A, we define its attentiveness as: the task of the agent is to proceed the quiz game smoothly. The agents utters for that purpose at the timings when the participants do not feel annoying and are likely to listen to. In order to improve the effectiveness of the agent's utterances which are expected to affect the participants, the addressee of those utterances is the participant who is most likely to have influences on the other participants. The personality of the quiz agent is neutral, i.e. do not try to help the participants and do not try to confuse the participants, either.

The following sections describe the central parts for realizing attentive quiz agent A, an

attentive utterance policy, the method that estimate the participants' status, and the implementation of attentive quiz agent A.

5.3.1 Attentive Utterance Policy

Attentive quiz agent A's utterance policy is designed based on the following principles:

- Prevent to be thought annoying, do not talk to the participants when they are actively discussing.
- If the participants do not answer the quiz for long time, the agent tries to urge the participants to answer or to press the hint button.
- In order to keep the quiz game active, the agent tries to stimulate the conversations of the participants by talking to them.
- When an utterance done by the agent is expected to stimulate some reactions from the participants, the person who is most likely leading the conversations of the participants is chosen as the addressee.

Considering the seven phases of the quiz game mentioned in section 5.1, two situations are considered as most unnatural. First, during the period after the agent issues the quiz question and before the participants answer it, the agent just stand there without doing anything. Second, the agent issues next quiz directly after the comment about the answer of current question. The utterance policy is then designed in addressing these two situations according to the activity of the conversation of the participants.

After issuing a quiz and before the participants answer it: If the participants keep interacting with each other actively, the agent does nothing. If the activity is initially high but becomes low later, in order to make the quiz game progress and stimulate the activity of the participants, the agent urges the participants to answer or reminds them the availability of hint (*Urge utterances* hereafter). However, because Urge utterances have to be designed depending on the quiz question, the variations are limited. They are uttered by the agent for at most twice in the period of one quiz. If the interactions among the participants are never active, Urge utterances are triggered by a 50-second timer. The relationships between time,

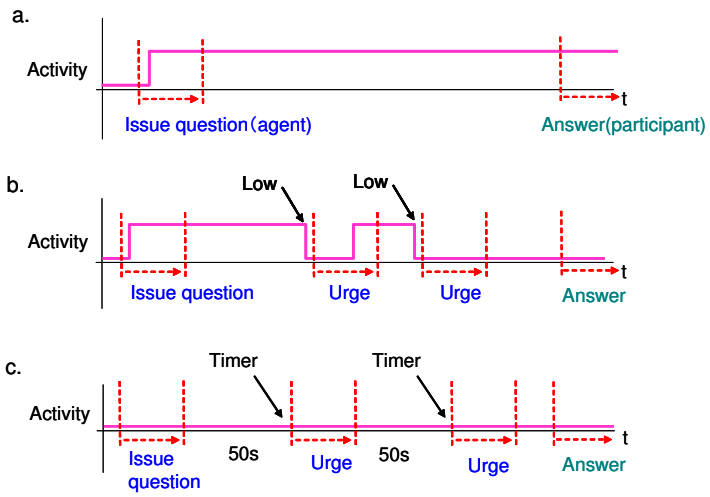

Figure 5.4: Utterance policy: after quiz issuing. (a) The activity is always high. (b) The activity is high at first but becomes low. (c) The activity never becomes high

participants' activity and the behaviors of the agent are shown in Figure 5.4. Since when the agent urges the participants, the reactions (press the hint button or answer the quiz) from them are expected, the addressee of Urge utterances are set to be the participant who are leading the group at that time.

After announcing the answer and before next quiz: If the activity of the participants become low while the agent is announcing the answer, the agent makes comments about the answer, cheers up or praise the participants (*Comment utterances* hereafter). If when the answer announcement ends, the participants are actively conversing, the agent suspends issuing next quiz or the final summary (*Proceed utterances* hereafter) until the participants calm down. The relationships between time, participant activity and the behaviors of the agent are shown in Figure 5.5. Figure 5.6 shows an example of how the policy is being executed during the subject experiment that will be discussed in section 5.5.

5.3.2 Participant Status Estimation

In order to take out the utterance policy described in last section, it is necessary to measure how active the participants' conversation is and who is the person leading the conversations in the group. We then define the two heuristics, *Interaction Activity (AT)* and *Conversation*

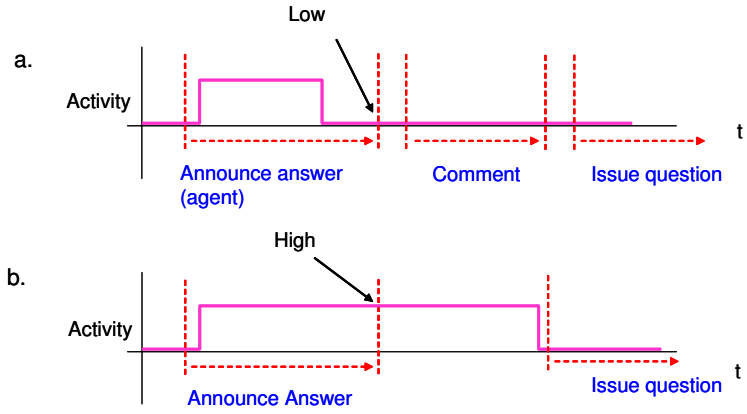

Figure 5.5: Utterance policy: after answer announcement. (a) The activity is low when the answer announcement ends. (b) The activity is high when the announcement ends

Leading Person (CLP) as follows:

Interaction Activity (AT): It indicates whether the users are active in their interactions. *High* and *low* are the two possible measured status. AT is high when all of the members of the participant group reacted to an utterance done by one of them with successive utterances and intensive face movements. AT is low otherwise.

Conversation Leading Person (CLP): It is the participant who is most likely leading the group at certain time point. It is estimated by counting who spoke at most and initiated most AT status of the group.

The computation of AT and CLP is reset at the beginning of each quiz based the assumption that the participants' activity heavily depends on the quiz. The intensity of face movements is approximated from the face orientation information measured by a WebCam and Omron's OkaoVision (Omron Corp., 2008) face detection library. C_t that means how much each participant paid attention to the screen at certain time point t is computed from N sampling data by the following equation.

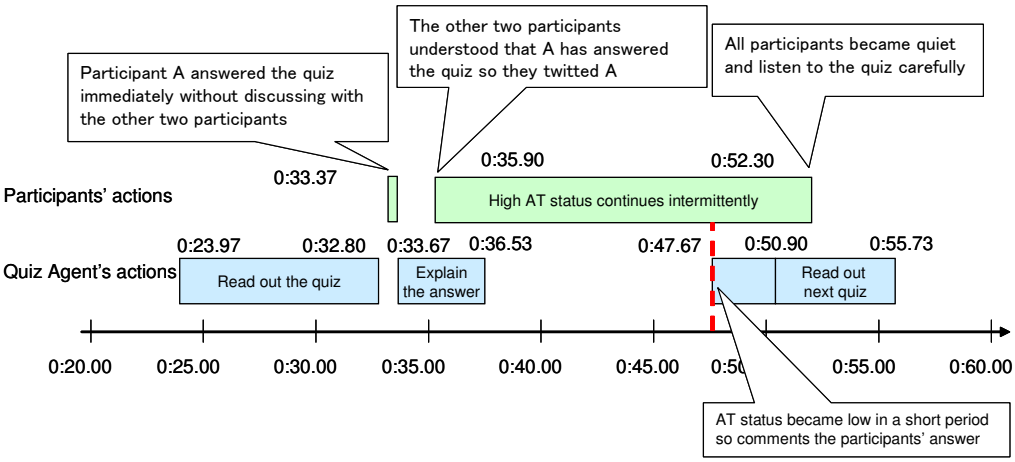

Figure 5.6: One example of how attentive quiz agent A's utterance policy works

$$
\begin{aligned}
V(t) &= (x_t, y_t) \qquad here \; -\frac{\pi}{2} \le x_t,\, y_t \le \frac{\pi}{2} \\
V_{max} &= (x_{max}, y_{max}) \\
f(V(t)) &= \left\{ \begin{array}{ll} 1 & if \quad -V_{max} \le V(t) \le V_{max} \\ 0 & if \quad -V_{max} > V(t) \quad or \quad V_{max} < V(t) \end{array} \right\} \\
C_t &= \frac{\sum_{k=0}^{N} [(N-k)^2 \times f(V(t-k))]}{\sum_{k=0}^{N} (N-k)^2} \qquad here \quad t \ge N
\end{aligned}
\tag{5.1}
$$

Here, $V(t)$ is the face orientation of a participant at time t (0 when the direction is toward the camera), while x_t and y_t represent the angle in horizontal and vertical directions within the range $\pm\pi/2$. V_{max} is the threshold to judge whether the participant is looking at the screen at t (the angles in horizontal and vertical directions: x_{max} and y_{max}). $f((V(t))$ denotes whether the participant is looking at the screen, $f((V(t)) = 1$ when (s)he is looking at the screen and $f((V(t)) = 0$ otherwise. When C_t is lower than the value α, this participant is regarded as not paying attention to the screen (the agent) and is having intensive face movements.

These parameters are conducted with the assumption to use the system in the experiment space shown in Figure 5.7, the number of participants is fixed to be three. Because the width

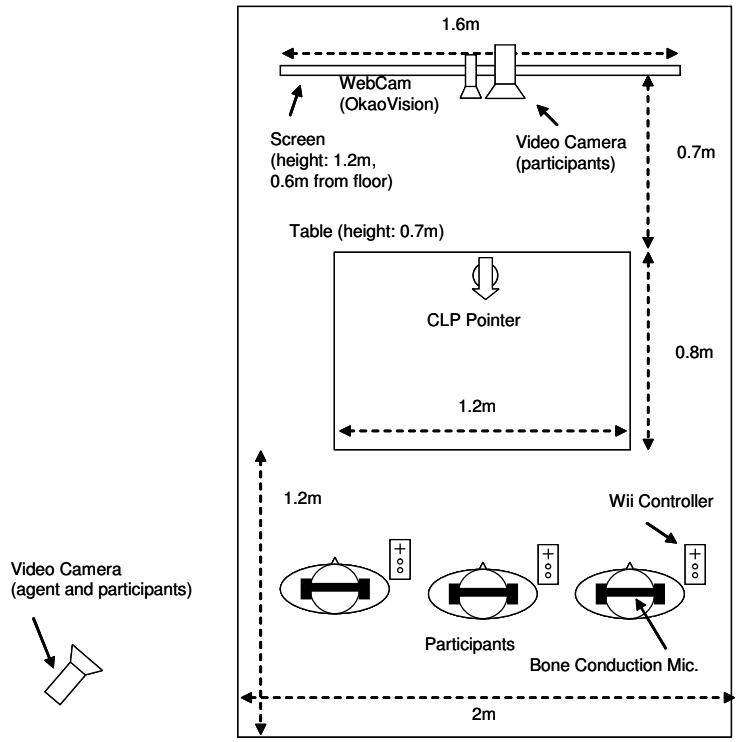

Figure 5.7: The experiment space of attentive quiz agent A

of the screen is nearly the same as the width of the whole space, and its height (1.8 m)is assumed to be higher than most Japanese participants, the participants are assumed to face orthogonal to the screen when they are looking at it. Therefore, x_{max} and y_{max} are set to be the middle of 0 and $\pm\pi/2$, that is, $\pm\pi/4$ to distinguish the directions of the screen and the other participants. The other parameters are conducted by empiric results. When $N = 12$ and $\alpha = 0.7$, appropriate results could be gotten.

On the other hand, with the presumption to port the system easier to real-world exhibitions, speech recognition is not used because it is too sensitive. Whether the participants are speaking or are in a conversation is detected only with acoustic information. A 2-second silent period is used to detect speaking segments from the voice streams of the microphone attached on each participant. The information is combined from all participants to detect whether a conversation is existing if their successive utterances do not break longer than two seconds. A conversation sequence is judged to be in high AT status if anyone of the

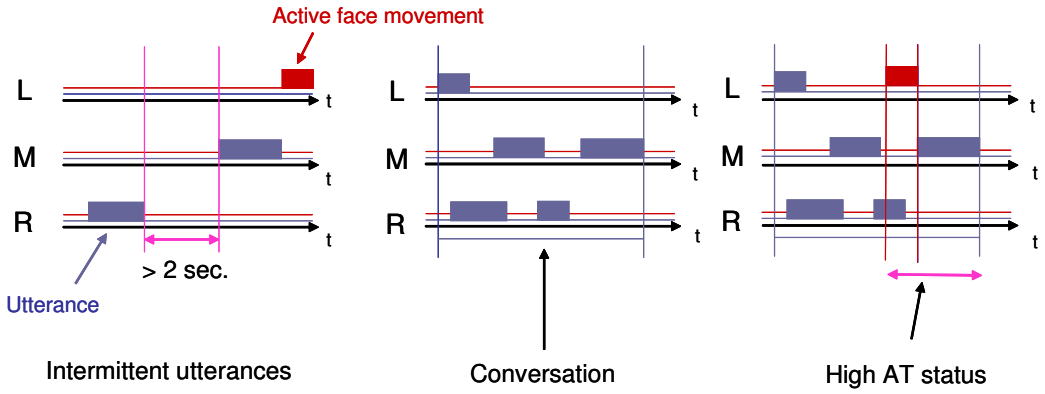

Figure 5.8: The criteria to judge a conversation sequence and high AT status. “L,” “M,” and “R” denote the three different participants

participants has active face movements (Figure 5.8). The changing AT status is used to further partition the conversation segments, the participant who is the starting point of each AT period is counted to initiate AT status once.

CLP is then estimated by tracking how many times each user spoke, and how many times he or she initiated an AT status of the participant group. Each participant is ranked according to these two criteria. The participant who spoke most is assigned with three points while who spoke least is assigned with one point. The participant who initiated most AT is assigned three points and who initiated least AT is assigned one point. These two scores are then summed with the same weight, the participant who has most points is judged as the CLP at that moment. The system constantly computes the CLP and thus there is always one CLP at any time point. There may be some periods when all of the participants are not speaking but are paying attention to the system. We assume that even there is no conversation in progress, the participants should be thinking about the answer based on their last conversation which should be counted as being influenced by last CLP participant. In the other words, we assume that even in a quiet period, there is a CLP participant (last one).

5.3.3 Implementation

Attentive quiz agent A is implemented in GECA framework, too. System functionalities are distributed into concurrently running components that are connected in the topology shown

in Figure 5.9. As shown in Figure 5.10, each participant is equipped with a Nintendo Wii remote controller, so that everyone can answer the quiz directly without the constraints of the distance to the touch panel that may have influences on the computation of CLP. Each one of them is also equipped with a bone conduction microphone to prevent the voice from the other participants to be collected. Due to the "Mona Lisa Effect" of 2D agents what was discussed in Chapter 4, the users can not correctly perceive the gaze direction of the agent except the middle one. A physical pointer is therefore introduced for the quiz agent to show who is the addressee of its utterances.

Each microphone is connected to an *Audio Processing* component that digitalizes the voice, extracts the sounds within human voice frequency range, and determines whether that user is speaking from the volume. The *Conversational Status Detection* component judges whether there is a conversation existing among the participants via the overlapping and successive relationship between the participants' utterances. A 2-second silence border is used as the threshold to distinguish two segments.

Video information taken by a WebCam is processed by the *Video Processing* component mainly utilizes OkaoVision face detection library. Recognized face orientations of the users are sent to the *Input Understanding* component for further processing. Because the OkaoVision library fails to recognize faces outside its range ($\pi/3$ in horizontal direction and $\pi/6$ in vertical direction), to compensate this and enumerate the jitters, Cam Shift method in OpenCV (Intel Corp., 2006) and Kalman filter are applied. The face direction is recognized at around 4fps on the computer used by us.

The face movement intensity information and the conversation status information is then combined by the *Input Understanding* component to estimate AT and CLP. Current AT and CLP are used to judge when to do what to whom by the *Dialog Manager* component, animation commands are then generated by it to drive the *Character Animator* component to render CG character animations.

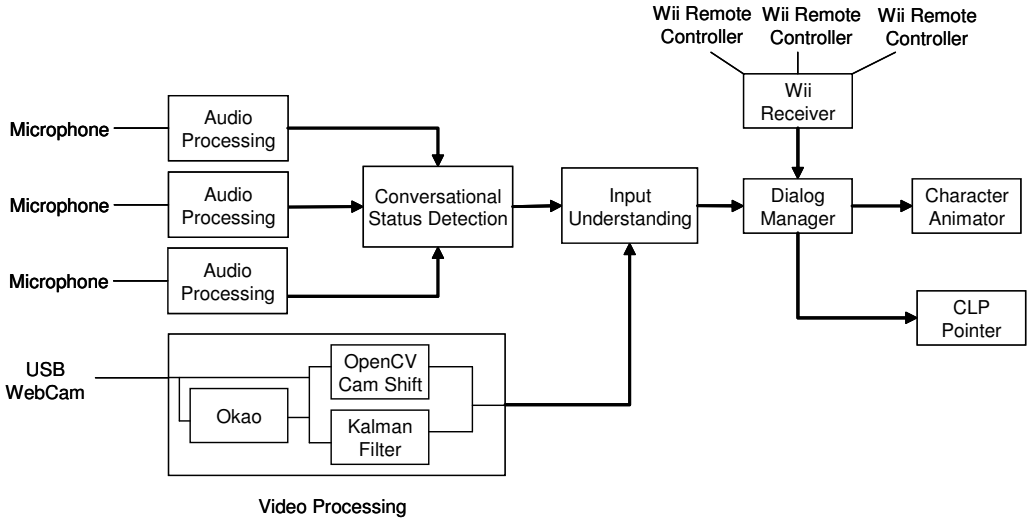

Figure 5.9: The system architecture of attentive quiz agent A

5.4 Quiz Agent with Attentive Attitudes toward Multiple Participants (Agent B)

Doing *idle motions* when the agent is in its stand-by status is one of the essential factors for ECAs in the sense of life-likeness (Egges & Visser, 2004). We humans can never keep still for a period, we do eye blinking, change postures because of the fatigue of legs, etc. For ECAs, these subtle animations are usually realized by replaying prerecorded animation sequences from motion capture data of real humans (Egges & Molet, 2004). However, these canned animations are fixed can can not be meaningfully adapted according to the status of the participants. Some other works attempted to realize the feedback behaviors that plays important roles in smooth conversations. Rapport Agent (Gratch et al., 2006) do listener feedback behaviors in responding to acoustic characteristics of the user's speech. Max (Kopp, Bergmann, & Wachsmuch, 2008) does real-time feedback behaviors when the user is typing with a keyboard. Both of these two works are realized with a predefined rule set. In Mack (Y. I. Nakano et al., 2003), the authors achieved natural nonverbal grounding via a statistical model conducted from the results of Wizard-of-Oz (WOZ) experiments.

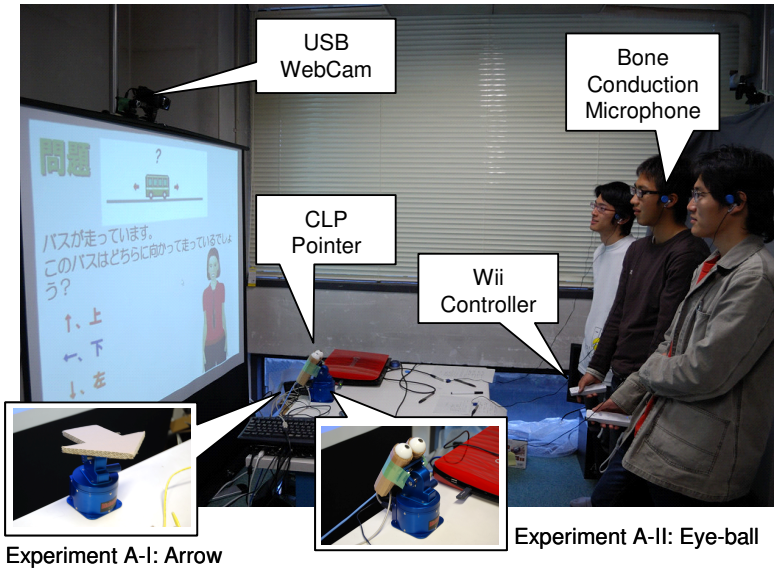

Figure 5.10: The sensor device configuration of attentive quiz agent A

Realizing observable nonverbal behaviors that expressing the quiz agent's internal attitude toward the participant group is sophisticated. The temporal granularity of the interactions is in milliseconds. The rules are also not clearly discovered yet personality dependent. If those rules existed, the size of the rule set is supposed to be huge to describe every possibility. Therefore, machine learning methodology is adopted in attentive quiz agent B. Support Vector Machine (SVM) classifier is chosen because its stability in achieving high accuracy.

5.4.1 The State Transition Model of the Attitude of Attentive Quiz Agent B

Considering the task to be a quizmaster waiting for the participants to answer the quizzes, the agent can be considered natural to have the attitudes ranging from *anxious*, *calm* to *impatient* toward the participants. In realizing attentive quiz agent B, these attitudes are defined as follows.

Calm: The agent feels that it should not disturb the participants but just keep concerning about them. The typical situation is: the participants are paying attention to the quiz

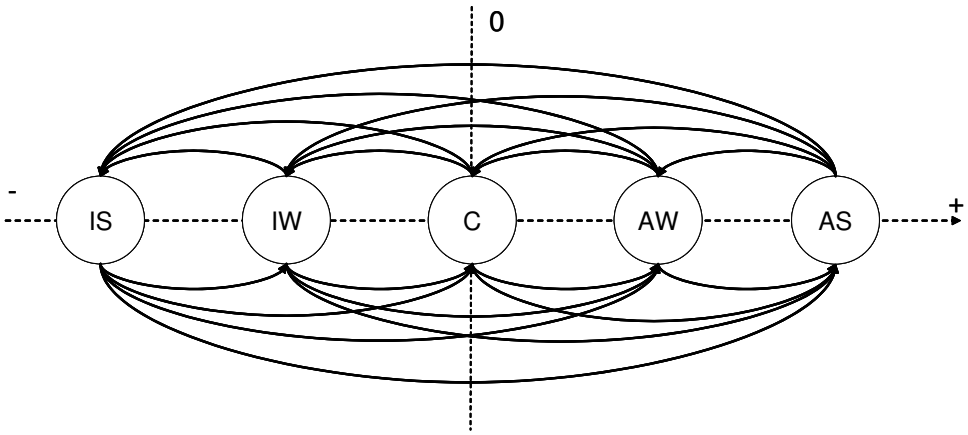

Figure 5.11: The five state of attentive quiz agent B's internal attitude at the axis with positive and negative direction toward the participants

game (the screen), and their discussion is active while the time past is not long.

Anxious: The agent feels anxious about the participants because them seem to have problem in answering the quiz. The typical situation is: the participants are paying attention to the screen, but their discussion of the answer is not active. If the attitude becomes stronger, the agent may try to affect the participants by telling the availability of hint.

Impatient: The agent starts to feel impatient about the participants. The typical situation is: the participants are actively discussing the answer and seem to have ignored the existence of the agent after a fairly long time since the agent issued the question. If this attitude becomes stronger, the agent may try to urge the participants to answer the quiz.

Anxious and Impatient states are further divided into *weak* and *strong* and therefore formed a five-state attitude model (C, AW, AS, IW, IS, Figure 5.11) of the quiz agent. Although the five states distribute on one axis, as the figure depicts, the agent's attitude may transit from one state to any other four states directly.

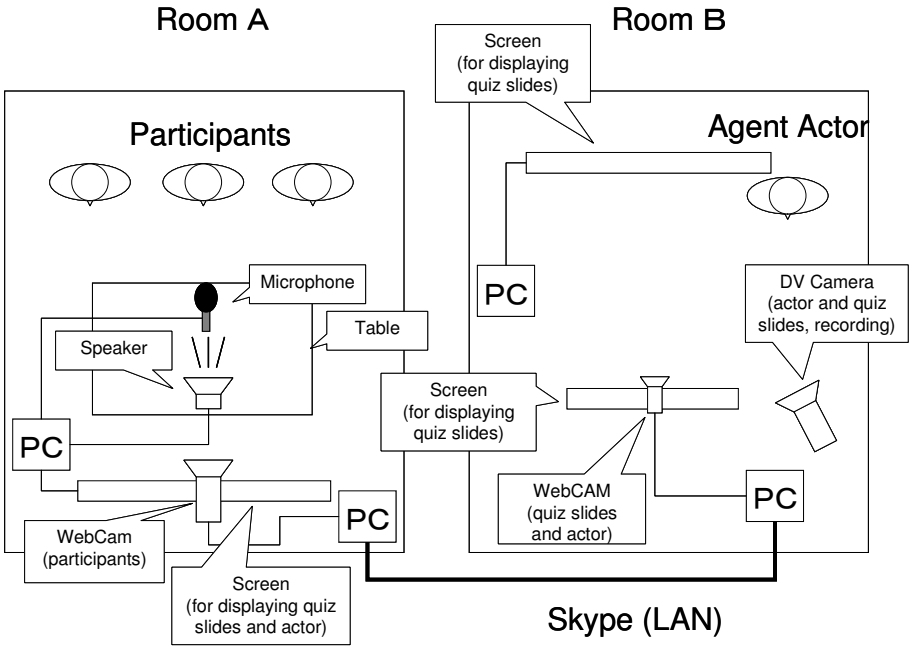

Figure 5.12: The settings of the WOZ experiment for state transition learning of attentive quiz agent B

5.4.2 The Acquisition of State Transition Rules

To acquire the training data for the SVM classifier, a WOZ experiment with two three-people groups is conducted at first. Instead of the CG character, one actor as the agent in room B is shown on the screen of room A and interacts with the participants who are in room A in real-time (Figure 5.12). The process is recorded by two cameras while this experiment is in progress. This actor then annotated his own attitude during the experiment with the iCorpusStudio (Nakata et al., 2009) video annotation tool. Since in realizing attentive quiz agent B, only the situation when the agent is waiting for the participants to answer the quiz, only 8'16" (496 sec) of the 11'17" video corpus was annotated. The distribution of the length of each state is listed in Table 5.3.

Since the video corpus was taken with 30fps video cameras, totally 14,890 training data are extracted from the video corpus. They are fed to the SVM classifier for learning the transition rules among the five states. In addition to the state label, the following four criteria are used in the training of SVM classifier.

Table 5.3: The results after the labeling of the video corpus of the WOZ experiment

State	–	C	AW	AS	IW	IS
Duration (s)	171.2	217.2	75.8	75.9	66.2	60.7
Number of Labels	10	13	7	4	7	3
Duration Average (s)	17.1	16.7	10.8	19.0	9.5	20.2

Table 5.4: The two-class thresholds of the learned SVM classifier

Class	C-AW	C-AS	C-IW	C-IS	AW-AS
Threshold	−2.7593	2.0750	4.0175	−1.0446	−10.2751
Class	AW-IW	AW-IS	AS-IW	AS-IS	IW-IS
Threshold	−1.9772	−2.4783	11.1825	8.4473	−3.8470

- Averaged face orientation of the participants in past three seconds.
- Volume of the voice collected by single environment microphone.
- Category of the quiz. Quizzes about knowledge, quizzes require logical inference, or quizzes requires some tricks.
- Time past since the agent issued the quiz.

Here, the averaged face orientation is computed as: OkaoVision's face orientation output (assign the value 1 if this participant is facing to the screen) times the confidence output. By using radial basis function kernel, the accuracy 73.2% is achieved in 10-fold cross verification on the learned classifier. Because SVM is originally a method to classify data from two classes, it is extended to run $C_2^5 = 10$ times for classifying data to five classes. The threshold for each pair is shown in Table 5.4. Table 5.5 shows the number of support vectors of each class. The numbers of support vectors are relatively high, this means the bounders of the five classes are complex and it was difficult to classify the corpus data. Further tunings on the parameters to improve the classifier's performance in generalization may be a future work.

Table 5.5: The number of support vectors (SV) of each class

Class	C	AW	AS	IW	IS	Total
Num. of SV	2,062	2,246	1,190	1,757	883	8,138

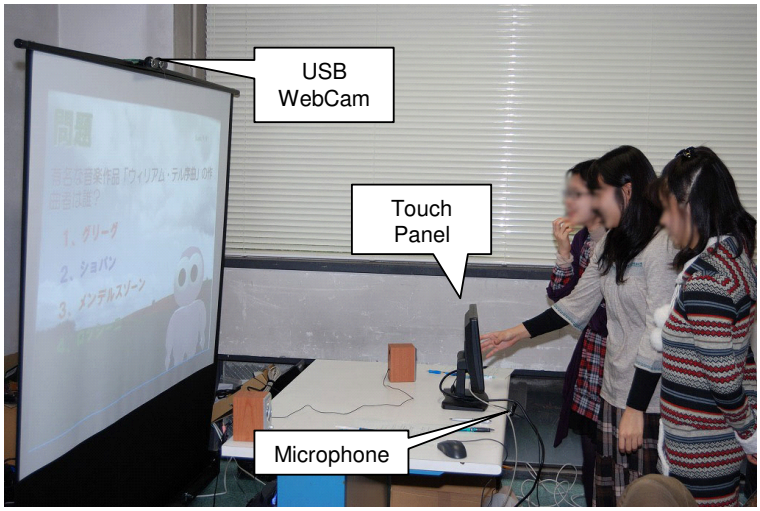

Figure 5.13: The hardware configuration of attentive quiz agent B

5.4.3 Implementation

The hardware configuration and experiment space configuration are shown in Figure 5.13 and Figure 5.14, respectively. The experiment space is basically the same as attentive quiz agent A except some changes on sensor devices. Contrary to attentive quiz agent A, the touch panel setting as the exhibited prototype is kept because CLP estimation is not used. The bone conduction microphones attached on individual participants are replaced by one single environment microphone, too.

Attentive quiz agent B is implemented in GECA framework, too. The components are shown in Figure 5.15. Benefits from GECA, the *Audio Processing*, *Video Processing*, *Touch Panel Controller*, and *Character Animator* components of previous systems are reused with some parameter tunings. Video and audio inputs from a WebCam and the environment microphone are fed to the *SVM* component that is implemented with LIBSVM (Chang & Lin,

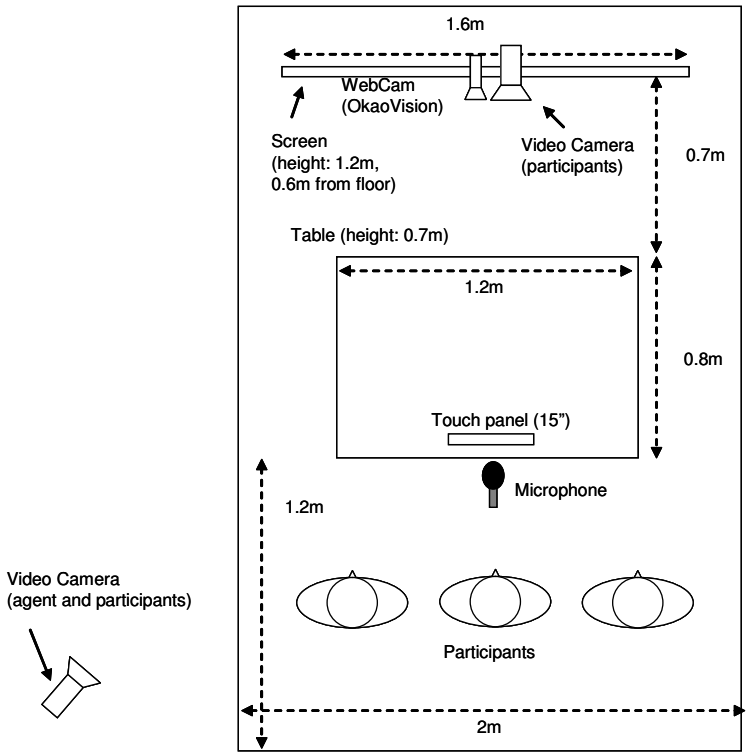

Figure 5.14: The experiment space of attentive quiz agent B

2008). The SVM component classifies current participants' status into one of the aforementioned five-state model. The *Dialog Manager* component that is implemented dedicated to this agent then repeatedly play corresponding animation sequences. In each one of the five states, the agent will then perform corresponding nonverbal animations in different strength. In Anxious-Strong (AS) state, the agent makes utterances like "Is the quiz difficult?" or "The hint of this quiz is available." In Impatient-Strong (IS) state, the agent makes utterances like "I think it's time to answer the quiz." or "How about the conclusion?" In Anxious-Weak, Impatient-Weak, and Calm states, however, the agent performs only nonverbal animations without making utterances. In order to prevent the agent from talking to the subjects too early due to the relatively unpredictable state transitions, utterances of the agent in first 15 seconds are suppressed. Also, the utterances are set to be at most three times.

In order to let the participants feel the agent's attitude more easily, instead of the female character used in the first NFRI quiz agent prototype and attentive quiz agent A, an abstract character called *Korosuke* is designed for attentive quiz agent B. Exaggerated nonverbal

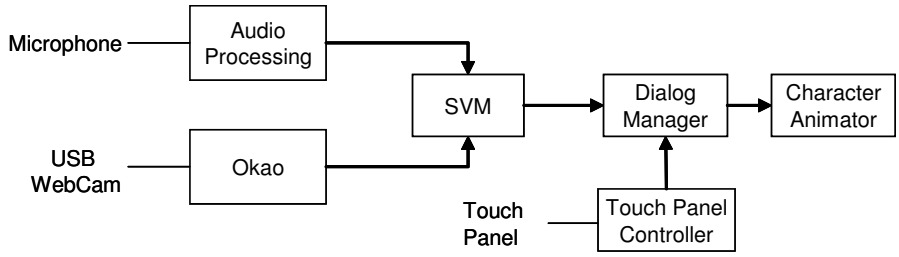

Figure 5.15: The system architecture of attentive quiz agent B

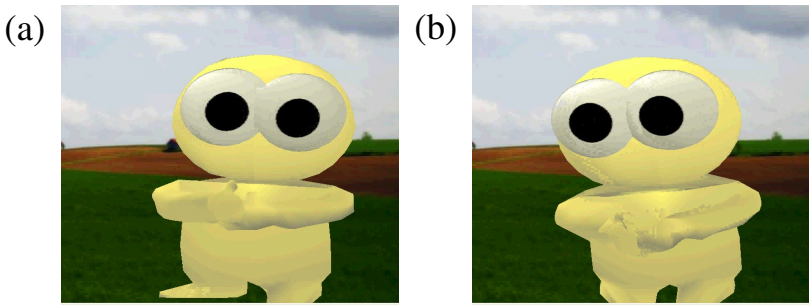

Figure 5.16: The nonverbal behaviors of the Korosuke character used in agent B system (a) Korosuke is in his Impatient-strong state, he folds his arms before his chest and beats his feet on the ground (b) Korosuke is in his Anxious-strong state, he bends his upper body forward, moves his head to look around to show the concern of the participants

behavior animations that express the five attitude states are then specially designed for the Korosuke character (Figure 5.16).

5.5 Evaluation Experiments

In the ECA research field, the usual research goal is to achieve human likeness that is an internal feeling and can not be objectively measured by an instrument. ECA researchers usually used questionnaire evaluation up to now. However, questionnaire investigation is considered to be not reliable, not objective and not a scaled measurement. In addition to the regular questionnaires, we adopt GNAT that is one of quantitative psychology methods to evaluate the subjects' implicit impressions toward agent A and agent B. In order to have a deep insight on how the participants reacted to the attentive agents, video analysis on the

video data collected during the experiments are done as well.

5.5.1 The Go/No-go Association Task (GNAT)

GNAT (Nosek & Banaji, 2001) test is a method indexing an implicit attitude or belief by assessing the strength of association between a target category and two poles of an attribute dimension. It is based on Signal Detection Theory (SDT) and the hypothesis that humans' accuracy in discriminating certain concept (category) and items of an attribute from distracters ought to be higher than the accuracy in discriminating that category and opposite items from distracters. The difference in accuracy (or sensitivity in SDT's terminology) between these conditions is taken as a measure of automatic attitude.

The test procedure is taken as follows. The test category term (e.g. character A) and the test attribute (e.g. natural) is shown at the left-upper and the right-upper corners of the program window, respectively (Figure 5.17). The subject has to categorize a word or a picture (stimuli) shown at the center coincides to either the category or the attribute. The stimulus may be targets (signal, e.g. human-like) or distracters (noise, e.g. artificial) and are only shown on the screen for a very short interval (response deadline, usually from 500 to 1,000 ms). If the subject's judgment is positive, then (s)he has to press the space key within the response deadline (Go) or do nothing otherwise (No-go). If the judgment is correct, a green "O" will be shown at the bottom of the screen and a red "X" will be shown otherwise during the time between two trials (inter-stimulus interval, ISI). Practice trials are conducted for the subjects to learn the correct categorizations. A correct "Go" is called a "hit" and an incorrect "Go" is called a "false alarm." The sensitivity d' of a subject regarding the test category and the test attribute is defined as $d' = Z(h) - Z(f)$, where h is the ratio of the number of hits over all signal trials and f is the ratio of the number of false alarms over all noise trials.

$$\begin{aligned} d' &= Z(h) - Z(f) \qquad where \\ h &= \frac{hits}{total\ \ number\ \ of\ \ signal\ \ trials} \\ f &= \frac{false-alarms}{total\ \ number\ \ of\ \ noise\ \ trials} \end{aligned} \tag{5.2}$$

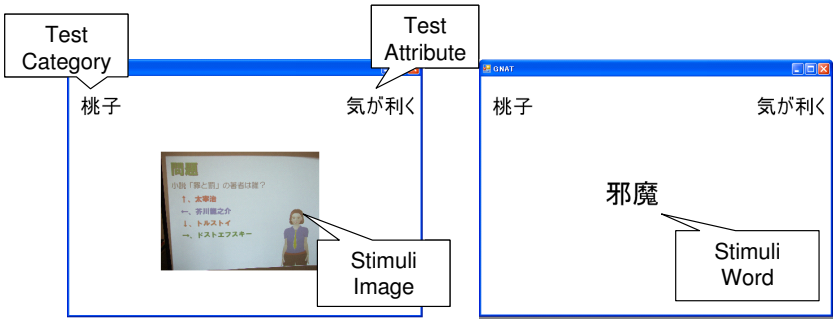

Figure 5.17: The screen-shot of our GNAT test program

From the definition of d', we can know that it is a measure of the distance (unit: standard deviation) of the probabilistic distribution for the user to have positive reaction to a signal stimuli from the normalized noise distribution. In the cases where the value is 0, the subject is assumed to press the key randomly while if the value is negative, it means that the subject misunderstood the procedure of GNAT. Therefore, only positive values should be considered in GNAT results. We adopted GNAT as a suitable evaluation for embodied conversational agents because its two good characteristics. The evidence for the signal drawn from the the stimulus can be presented by a single numeric value. Z score is used in the computation, so that the results from different subjects can be compared in scaled methods.

5.5.2 Common Experiment Settings

The experiment participants are recruited in the university campus with only one prerequisite that they must enroll as three-people groups. The participant groups are then assigned to the evaluation experiments randomly. Each group played quiz game with agent A or agent B for one session and their compared system for the other session. In order to make active conversations among the participants more expectable, they are instructed that the reward varies according to their performance in the game. To achieve counterbalance, the order of the internal algorithms, the external appearance (color or clothes), and the quiz contents of the agents and the session order are switched every session (Table 5.6). Since there are three changing factors in this case, eight groups of participants are required in each experiment. A questionnaire investigation is taken immediately after each session, and the GNAT test is

Table 5.6: An example schedule with counter-balance in the experiment for evaluating attentive quiz agent A. In this experiment, it is compared to another quiz agent with fixed-timings on utterances

	1st session			2nd session		
Group	Clothes	Quiz Set	Algorithm	Clothes	Quiz Set	Algorithm
1	Blue	1	Fixed	Red	2	Attentive
2	Red	1	Fixed	Blue	2	Attentive
3	Blue	2	Fixed	Red	1	Attentive
4	Red	2	Fixed	Blue	1	Attentive
5	Blue	1	Attentive	Red	2	Fixed
6	Red	1	Attentive	Blue	2	Fixed
7	Blue	2	Attentive	Red	1	Fixed
8	Red	2	Attentive	Blue	1	Fixed

taken after the two experiment sessions.

The settings of GNAT test used in the experiments are as follows:

- Response deadline is 600 ms.
- Because ECAs are not supposed to be common sense, ISI is set to relatively longer 1,500 ms.
- 20 practice trials for target category and target attribute each. 40 critical (test) trials are conducted in each being evaluated agent.
- The ratio of signal and noise stimulus is 1:1 and all of them are in the same category (see Table 5.8 and 5.18).

5.5.3 The Evaluation of Attentive Quiz Agent A

Considering the functionalities of CLP pointer, to attract the participants' attention and to indicate the addressee of the agent's utterances. The shape of the pointer can be considered to have great influences on the participants' reactions. Therefore, in the evaluation experiment of attentive quiz agent A, two shapes of CLP pointer are adopted. One of them is simply an arrow, but the other one has two ping-pong balls marked with black dots on its

Table 5.7: The different settings of attentive quiz agent A (Attentive) and fixed timing quiz agent (Fixed) in experiment A-I/II

Utterance Timing	Attentive	Fixed
Urge	utterance policy	every 50 seconds
Comment	utterance policy	immediately after answer announcement
Proceed	utterance policy	immediately after comment utterance
CLP Pointer Addressee	**Attentive**	**Fixed**
Urge	CLP	random
Otherwise	upward	upward

top (eye-ball hereafter, Figure 5.10). They are investigated in two experiments, A-I with arrow pointer and A-II with eye-ball pointer, respectively. Eight groups (average age 21.3, 18 males and 6 females) of participants are chosen randomly to attend experiment A-I, another eight groups (average 21.9, 21 males and 3 females) attended experiment A-II.

In each experiment, attentive quiz agent A is compared with the other agent called fixed timing agent. It is exactly the same as attentive quiz agent A except the utterance timings are fixed and the addressee of CLP pointer is randomly decided. The relationship between the 2D graphical agent character and the physical CLP pointer is not explicitly specified in the instruction, but the participants are instructed that when the pointer is pointing at one of them, it means that the 2D character is only talking to that person, and while the pointer is pointing upward, that means the 2D character is talking to all of them. The details of differences between attentive quiz agent A and fixed timing agent is shown in Table 5.7.

GNAT Test

The GNAT stimulus of experiment A-I and A-II are shown in Table 5.8 for the being tested attribute, "attentive." The valid GNAT results are shown in Table 5.9. In these two experiments, the results of GNAT test were stable and similar. The difference between agent A and fixed-timing agent was not significant both in experiment A-I (t test: p=.57, two tailed if not mentioned hereafter) and A-II (t test: p=.22). But agent A was more associated with *attentive* by more participants in both experiments (11:8 in experiment A-I and 12:8 in experiment A-II). In order to see whether the shape of CLP pointer has influences on the

Table 5.8: The stimuli used in the GNAT test of experiment A-I/II. The terms coincide to the test category, "attentive" are chosen as signals and the opposite term are chosen as noise

Signal	Noise
気配り (attentive)	鬱陶しい (annoying)
心配り (considered)	進行ベタ (clumsy)
円満 (harmonious)	邪魔 (disturbing)
和やか (genial)	お節介 (officious)
仕切り上手 (competent)	横やり (interruption)
名司会 (smooth)	自分勝手 (selfish)
控えめ (moderate)	わがまま (willful)
適切 (appropriate)	でしゃばり (meddlesome)
丁度いい (just)	過剰 (excess)
テキパキ (efficient)	独りよがり (opinionated)

attribute, "attentive," t test is applied to the results of attentive quiz agent A of experiment A-I and A-II. The test result showed that the two groups can be considered as the same (t test: p=.93), therefore, we can conclude that shape of CLP pointer does not have influence on the participants' implicit attitude toward the concept, "attentive."

Questionnaires

The results of questionnaire investigation and Wilcoxon signed-rank test are shown in Table 5.10 and 5.11. The results of Mann-Whitney U test on the questionnaires of attentive quiz agent in experiment A-I and A-II are shown in Table 5.12.

In both experiments, the participants paid more attentions on the movements of agent A's CLP pointer (Q10, A-I: p=.08, p<.01). This shows that the participants are conscious the different meanings of the pointer's indication between attentive agent A and fixed timing agent. Moreover, in both experiments, the participants felt uncomfortable about the CLP pointer (Q12, A-I: p=.20, A-II: p=.02) especially in experiment A-II. This can be considered because the shape of eye-ball is too offensive so that the participants felt that they were being looked at by somebody despite it attracts more attention. It seems because the same reason, the eye-ball pointer is more comprehensive (Q11, U test, p=.08), participants themselves paid more attention to the pointers and thus felt that the agent paid more attention to them (Q8, A-II: p=.09, U test: p=.03).

Table 5.9: The valid GNAT test results of experiment A-I/II. "F" denotes the fixed timing quiz agent and "A" denotes the attentive quiz agent

Ex. A-I				Ex. A-II			
ID	F	A	Att.	ID	F	A	Att.
1	0.511	1.095	A	25	1.199	2.073	A
2	1.806	1.407	F	26	0.549	0.800	A
3	0.260	0.245	F	27	0.639	0.511	F
5	0.588	1.366	A	28	2.073	0.778	F
6	0.289	0.651	A	29	0.928	0.778	F
8	1.227	0.967	F	30	0.639	0.842	A
10	1.392	1.199	F	32	1.227	1.422	A
11	0.842	1.519	A	33	0.771	0.674	F
12	1.282	0.253	F	34	0.385	1.028	A
13	0.511	1.060	A	35	1.290	0.650	F
15	1.095	1.227	A	37	1.156	1.036	F
16	0.379	1.049	A	38	0.651	0.549	F
18	1.645	0.456	F	39	0.126	1.227	A
19	1.516	1.878	A	40	0.674	1.422	A
20	0.896	1.049	A	41	1.683	2.030	A
21	2.926	2.486	F	43	0.910	1.366	A
22	1.036	1.366	A	44	1.120	1.260	A
23	1.036	1.422	A	45	1.683	2.486	A
24	1.227	1.156	F	46	1.282	0.524	F
				47	0.253	1.227	A
Avg.	1.077	1.150			0.962	1.134	
SD	0.624	0.519			0.490	0.538	

About the questions related to utterance timings, significant differences between attentive quiz agent A and fixed timing agent could not be found. In Q9, "The progress of the game was smooth (A-I: p=.11, A-II: p=.08)," the participants tended to feel that the game was not smooth with attentive quiz agent A. Since the fixed timing quiz agent always makes comments immediately after it announces the correct answer and immediately proceeds to next quiz without waiting for the participants to calm down from their active discussions, this may cause the participants a *faster* impression of fixed timing quiz agent. If the participants mistakenly interpret the meaning of *smooth* to *fast*, it could lead them an impression of the attentive quiz agent as *not smooth*. Because the objective of attentive quiz agent A's

utterance policy is not to make the quiz game progress *faster*, this may not be considered as a failure.

On the other hand, in the questions: Q5, “The discussion was active (A-I: p=.86, A-II: p=.07),” Q13, “There were silent periods in the session (A-I: p=.05, A-II: p=.68),” the results of attentive quiz agent are shown to shift in the positive direction from experiment A-I to A-II. Therefore, we can conclude that the eye-ball CLP pointer seems to stimulate the participants’ conversation more successfully.

Table 5.10: The questionnaire investigation results of experiment A-I. The questionnaires are evaluated in 7-scale where 1 means smallest degree and 7 means the highest degree. The results are then tested with Wilcoxon signed-rank test. M_F and IQD_F are the median and the inter-quartile deviation (IQD) of the fixed timing quiz agent. M_A and IQD_A are the median and the IQD of the attentive quiz agent A (arrow CLP pointer). "+," "-," and "0" columns mean the number of participants who evaluated attentive quiz agent with higher than, lower than or the same score as the compared fixed timing quiz agent respectively in each question. "+ rank" and "- rank" shows the mean ranks of positive and negative answers, the larger the number shows higher difference than the opposite answer. "p" shows the two-tailed probabilities of the statistical test

Q	Question	M_F	IQD_F	M_A	IQD_A	+	-	0	+ rank	- rank	p
1	The character was friendly.	4.0	1.00	4.0	1.50	6	8	10	5.42	9.06	0.199
2	The character's utterances were annoying.	4.0	2.00	3.0	1.25	5	8	11	5.10	8.19	0.159
3	The character was passive.	2.0	1.00	2.0	1.00	10	5	9	7.90	8.20	0.268
4	The character's behaved in responding to our status.	4.5	1.13	4.0	1.00	6	10	8	9.00	8.20	0.463
5	The discussion was active.	6.0	1.00	6.0	0.63	5	6	13	7.00	5.17	0.855
6	I often considered alone.	2.0	1.00	2.0	1.50	7	4	13	6.86	4.50	0.177
7	The character's behaviors stimulated our discussion.	4.5	1.13	5.0	0.50	8	7	9	6.94	9.21	0.793
8	The character paid attention to us.	3.0	1.00	3.5	1.00	8	8	8	9.12	7.88	0.792
9	The game progress was smooth.	5.0	1.13	4.5	2.00	4	11	9	8.00	8.00	0.109
10	I paid attention to the movement of the pointer.	2.5	2.00	4.0	2.00	12	4	8	8.50	8.50	**0.075**
11	The indication of the pointer was comprehensive.	2.0	1.50	2.0	1.13	4	6	14	6.38	4.92	0.835
12	The indication of the pointer was incongruous.	3.0	1.63	4.0	2.00	6	2	16	4.50	4.50	0.204
13	There were silent periods in the session.	2.0	1.50	3.0	2.50	10	3	11	7.30	6.00	**0.053**
14	I would like to response to the character's urges.	5.0	1.00	5.0	1.50	3	7	14	5.50	5.50	0.256

Table 5.11: The questionnaire investigation results of experiment A-II. The questionnaires are evaluated in 7-scale where 1 means smallest degree and 7 means the largest degree. The results are then tested with Wilcoxon signed-rank test. M_F and IQD_F are the median and the inter-quartile deviation (IQD) of the fixed timing quiz agent. M_A and IQD_A are the median and the IQD of the attentive quiz agent A' (eye-ball CLP pointer). "+," "-," and "0" columns mean the number of participants who evaluated attentive quiz agent with higher than, lower than or the same score as the compared fixed timing quiz agent respectively in each question. "+ rank" and "- rank" shows the mean ranks of positive and negative answers, the larger the number shows higher difference than the opposite answer. "p" shows the two-tailed probabilities of the statistical test

Q	Question	M_F	IQD_F	M_A	IQD_A	+	-	0	+ rank	- rank	p
1	The character was friendly.	4.0	1.00	4.0	1.00	7	8	9	8.43	7.62	0.954
2	The character's utterances were annoying.	4.0	1.50	3.5	2.00	9	8	7	8.50	9.56	1.000
3	The character was passive.	2.0	1.00	3.0	1.50	8	9	7	13.00	5.44	0.187
4	The character's behaved in responding to our status.	5.0	1.00	4.0	1.00	6	11	7	9.17	8.91	0.299
5	The discussion was active.	5.0	1.13	6.0	0.63	9	3	12	6.89	5.33	**0.068**
6	I often considered alone.	3.0	1.50	3.0	1.00	7	6	11	7.86	6.00	0.501
7	The character's behaviors stimulated our discussion.	5.0	1.63	5.0	1.50	10	6	8	8.40	8.67	0.403
8	The character paid attention to us.	3.0	1.50	4.5	1.00	11	3	10	7.23	8.50	**0.087**
9	The game progress was smooth.	5.0	1.13	4.0	1.00	6	14	4	9.92	10.75	**0.081**
10	I paid attention to the movement of the pointer.	2.0	1.63	5.0	0.88	14	3	7	10.07	4.00	**0.002**
11	The indication of the pointer was comprehensive.	2.0	1.00	3.0	1.00	11	3	10	7.73	6.67	**0.037**
12	The indication of the pointer was incongruous.	4.0	1.13	5.0	1.50	11	3	10	8.00	5.67	**0.024**
13	There were silent periods in the session.	4.0	1.50	4.0	1.50	6	8	10	7.67	7.38	0.678
14	I would like to response to the character's urges.	4.0	1.13	5.0	1.50	7	6	11	7.43	6.50	0.632

Table 5.12: The results of Mann-Whitney U test of the questionnaires of experiment A-I (arrow CLP pointer) and A-II (eye-ball CLP pointer). "Rank I" and "Rank II" denote the mean rank of experiment A-I and A-II results, respectively

Q	Question	Rank I	Rank II	p
1	The character was friendly.	23.25	25.75	0.531
2	The character's utterances were annoying.	23.25	25.75	0.531
3	The character was passive.	25.10	23.90	0.760
4	The character's behaved in responding to our status.	25.42	23.58	0.643
5	The discussion was active.	26.98	22.02	0.200
6	I considered alone.	24.54	24.46	0.983
7	The character's behaviors stimulated our discussion.	22.94	26.06	0.430
8	The character paid attention to us.	20.12	28.88	**0.027**
9	The game progress was smooth.	24.79	24.21	0.884
10	I paid attention to the movement of the pointer.	24.48	24.52	0.992
11	The indication of the pointer was comprehensive.	21.04	27.96	**0.080**
12	The indication of the pointer was incongruous.	23.42	25.58	0.588
13	There were silent periods in the session.	24.06	24.94	0.826
14	I would like to response to the character's urges.	24.77	24.23	0.892

Video Analysis

In order to have a deeper insight on how the attentive quiz agent affected the participants, analysis upon the video records collected during the experiment has been done. The video data are recorded from two cameras set up at the view points shown in Figure 5.7. For reducing the tendency caused by subjective judgment, four annotators who are familiar with video annotating but are not involved in the development of this study are asked to annotate the video data. The video data of two groups in experiment A-I and two groups in experiment A-II are selected randomly and are assigned to each annotator (eight sessions for every annotator). The video annotation tool, iCorpusStudio was used here, too. The objectives and the algorithms of this study were not included in the instructions for the annotators. The annotators are instructed to annotate the video data as the following conditions:

Utterance timings: for the purpose to see whether the agent utters at appropriate timings. The short periods when the agent just started to make *Proceed*, *Urge*, and *Comment* utterances are annotated. Since in either case, the first quiz is issued immediately after a long greeting, the situations when the agents are issuing first quizzes are not counted. The

following labels are available for timing annotations:

Smooth (S): nothing special happened, the quiz game proceed smoothly.

Abrupt (A): the agent talks to the participants at an abrupt timing, e.g. when they are in active conversation. The participants either ignored the agent's utterances and continued their conversation, or interrupted their current conversation suddenly and paid attention to the agent.

Tardy (T): the agent talked to the participants after the following situation, the system seem looked freeze, the participants look confused about why the game does not proceed.

Participants' attention: for investigating whether the participants paid attention to the agent's utterances. The periods during the agent is making *Urge* and *Comment* utterances are annotated. The short period just after the agent began to talk is ignored in this annotation. Since the *Proceed* utterances are relatively longer and are important to the participants, they always paid attention to the agent. Therefore, the Proceed utterances are not counted here. The following labels are defined for this annotation:

Listen (L): at least two participants are listening to the agent's utterance, or at least one of the participants replied to the agent, commented on the agent's utterance as well as other observable reactions to the agent.

Ignore (I): at least two participants are in their own conversation and are ignoring the agent's utterances.

Conversation Leading Person: when the CLP pointer is in action, whether the participant whom it is pointing is person who is leading the conversation of the group at that time point. If who is the CLP is not so clear at this point, then use the CLP of the whole session as the criterion. The following labels are defined:

Conversation Leading Person (C): the person pointed is the CLP at this time point.

Not Conversation Leading Person (NC): the person pointed is not the CLP at this time point.

Table 5.13: The comparison of the frequency of smooth utterance timings between attentive quiz agent A and the fixed timing quiz agent. The results are the combination from experiment A-I and A-II. The numbers without remarks represent the number of times

	Attentive Quiz Agent A			
	Proceed	Urge	Comment	Total
Smooth	112	45	67	224
Abrupt	42	17	35	94
Tardy	6	0	4	10
Smooth(%)	70.0	72.6	63.2	68.3
	Fixed Timing Quiz Agent			
	Proceed	Urge	Comment	Total
Smooth	104	14	78	196
Abrupt	47	10	73	130
Tardy	0	1	0	1
Smooth(%)	68.9	56.0	51.7	59.9

Unclear: the cases when the person pointed is not observable due to the view point of the camera and the activity of the participants. These cases are not counted in the analysis.

The comparison of utterance timings between attentive quiz agent A and fixed timing quiz agent is depicted in Table 5.13. According to the observation, there was nearly no difference between these two types of agent in making smooth utterances involving proceeding the game (P: 70.0%:68.9%). On the other hand, in the cases of Urge and Comment utterances, the attentive quiz agent tends to make smooth impression more often (U: 72.6%:56.0%, C: 63.2%:51.7%). The difference was particularly high in Urge utterances, this can be considered because the different properties of the two types of utterances. The total number is few, but attentive quiz agent caused the impression of tardy timings of utterances more often (10:1), this coincides to the results from the questionnaires.

The investigation on the influences of different combinations of utterance timings and types on the participants' attention is shown in Table 5.14. From these data, we can see that when the utterances are made at smooth timings, the participants tend to pay attention to the agent and listen to its utterances (C: 87.8%, U: 90.0%). In contrary to this, when the

Table 5.14: The influences of different combinations of utterance timings and types on the attention of the participants. "C" and "U" are the abbreviations of "Comment" and "Urge" utterance types. "L" and "I" represent the "Listen" and "Ignore" attention status of the participants. The results are composed from the ones from experiment A-I and A-II, and the numbers without remarks represent times

Attention	Smooth	Abrupt	Tardy
C-L	129	41	4
C-I	18	65	1
C-L(%)	87.8	38.7	80.0
U-L	54	17	1
U-I	6	11	0
U-L(%)	90.0	60.7	100.0
L(%)	88.4	43.3	83.3

utterances are made at abrupt timings, the possibility for the participants to stop their own conversations and listen to the agent becomes lower (C: 38.7%, U: 60.7%). The reason why Comment utterances are particularly ignored can be considered due to its less importance to the participants, because they often felt surprised about the answer if they were wrong and discussed about the answer by their own after the answer announcement.

The difference of how often the agent is ignored according to different shapes of CLP pointer is shown in Table 5.15. From this observation, the order of how strong the agent could attract the participants' attention was: eye-ball pointer > arrow pointer > no pointer. On the other hand, from the truth that the utterances made to the CLP are constantly less often ignored, it implies the hypothesis that talking to the CLP should be able to cause the group to react more easily was correct.

CLP Estimation

In order to measure the accuracy of the CLP estimation method, the question, "who lead our group's discussion during the game?" is also in the questionnaire. The annotators are also asked to judge which participant tended to lead the discussions during the whole session. These two results are compared with the estimation of the system in the sense of time ratio

Table 5.15: The influences on the participants' attention from different shapes of CLP pointer and whether the addressee is current CLP or not. "C" and "NC" means that the pointer pointed on the person who is the current CLP or not respectively. The numbers without remarks represent times. The data of "Comment"utterances without the movements of CLP pointer is listed for reference

	Arrow		Eye ball		none
	C	NC	C	NC	—
Ignore	4	4	1	4	84
Listen	21	11	13	16	174
Ignore (%)	16.0	25.0	7.1	20.0	32.6

of each participant in Table 5.16.

The candidates of *correct answers* should be either from the participants themselves or from the annotators, however, by comparing the estimation of the system to them, the coincidence were both around 50%. In addition to this, the comparison between the judgment of the participants and the annotators also had around 50% coincidence. These results imply the difficulty in judging who is leading the conversation during a relatively long time (the whole session) as well as the estimation done by the system can get similar level of accuracy as humans. On the other hand, although the social relationship among the participants can be considered to have great influences on their answers in the questionnaire, it was not clear how it affected the participants in this experiment.

Table 5.17 shows the accuracy of the CLP estimation evaluated by the annotators when the pointer is in action. The accuracy is higher than the estimation on the whole session (60.4%:50.0%). The reason can be considered as: for humans, it is relatively stable in judging the CLP in short periods, but for longer periods (e.g. the whole session), the dynamically changing discussion (CLP) caused the impressions ambiguous and thus the difficulty in CLP judgment.

Summary

By summarizing the experiment results, we can conclude as follows:

The heart of attentive quiz agent A, the attentive utterance policy could not make the

Table 5.16: The comparison between the CLP from the estimation of the system, the judgment of the annotators, and the questionnaires answered by the participants themselves. The column ID denotes the 16 participant groups. The estimation results were shown as the percentage of time during the whole session when each participant is judged as the CLP by the system. "L," "M," and "R" mean the participant who stands at left, middle, and right positions respectively. As explained in section 5.3.2, the system always keeps the computation of CLP, so the percentages sum up to 100. "S," "A," and "Q" denote the judgement results from the system, the annotators, and the questionnaires filled by the participants respectively

ID	L	M	R	S	A	Q	S/A	S/Q	A/Q
1	0.3	51.7	47.0	M	M	M	○	○	○
2	20.1	54.0	25.7	M	L	R	×	×	×
3	4.8	32.9	62.0	R	M	R	×	○	×
4	74.5	12.9	8.6	L	L	L	○	○	○
5	44.5	12.9	42.2	L	R	M	×	×	×
6	21.3	30.9	47.4	R	L	L	×	○	×
7	9.6	32.0	57.9	R	M	R	×	○	×
8	4.2	15.4	80.1	R	L	L	×	×	○
9	56.4	33.0	10.6	L	L	—	○	—	—
10	34.6	56.5	8.9	M	M	L	○	×	×
11	0.1	5.2	94.8	R	R	M	○	×	×
12	73.9	17.4	8.7	L	L	L	○	○	○
13	16.2	25.4	58.4	R	M	M	×	×	○
14	4.1	2.4	93.5	R	M	M	×	×	○
15	1.1	17.8	81.1	R	R	R	○	○	○
16	20.6	40.6	38.8	M	M	M	○	○	○
						Coincidence (%)	50.0	53.3	53.3

participants to feel that the agent is *attentive*. Depending on the shape of the CLP pointer, it is possible to attract the participants' attention, stimulate their conversation to be active, but these behaviors do not cause an "attentive" impression.

In contrary to that, the hypotheses of the utterance policy and its required information, AT and CLP estimations could be considered partially successful. It is observed that if the agent talks to the appropriate participant (CLP) at an appropriate (smooth) timing, the utterance can be more expected to be effective (the participants listen to it). The evaluation of the AT estimation is difficult, however, from the fact that the attentive quiz agent A could make smooth utterance timing at higher percentages, the AT estimation method seems to

Table 5.17: The accuracy of CLP estimation judged by the annotators. “Attentive” is attentive quiz agent A and “Fixed” is the fixed timing agent

	Attentive	Fixed
CLP	32	10
Not CLP	21	12
Accuracy (%)	60.4	45.5

work properly.

The CLP estimation coincides to the judgment done by humans at the level from 50% to 60%. When the pointer is pointing at correct person (the CLP), then it can be expected that the participants will listen to the agent’s utterance. Despite the eye-ball CLP pointer is considerably more effective as a pointer device, the indication of its head like shape seems to be more offensive than an arrow pointer so that the participants felt more uncomfortable.

This implies that using a physical pointing device with the 2D agent can be an effective way to specify the addressee of the agent’s attention, but the utterance policy that always treats the person who is leading the conversation as the addressee may not be appropriate. Whom to point to and what to say at that time seem should to be more carefully and detailedly designed.

Although quantitative analysis could not be done, during the experiments, the reactions from the participants like saying “good work” or “yes, you are right”, or bow to the agent are often observed. These reactions can be considered as positive impressions to the agents, however, from the high ratio that the agent’s utterances were ignored by the participants (which should not happen in human-human conversations), we could not conclude that the agents are treated as life-like.

5.5.4 The Evaluation of Attentive Quiz Agent B

In order to evaluate attentive quiz agent B, it is compared with an agent whose internal attitude state transits randomly in experiment B. The behaviors corresponding to each attitude state are expressed the same bye these two agents. As part of the utterance policy of agent B, it does not speak in first 15 seconds to prevent to be felt as annoying, but the random agent

Table 5.18: The stimuli used in the GNAT test of experiment B. The terms coincide to the test category, "natural" are chosen as signals and the opposite term are chosen as noise

Signal	Noise
人間らしい (human-like)	ぎこちない (awkward)
気配りな (attentive)	ギクシャクした (jerky)
気を使った (thoughtful)	人工的な (artificial)
合理的な (reasonable)	機械的な (mechanical)
一貫した (consistent)	でたらめな (fake)
場に応じた (flexible)	変な (strange)
適当な (adequate)	妙な (weird)

does not have this limitation. Eight groups (average age: 22.5, 18 males and 6 females) participated in experiment B.

GNAT Test

In the GNAT part of this experiment, attentive quiz agent B is tested with the attribute, *natural*. The stimulus chosen to be suitable for describing naturalness is listed in Table 5.18. The participants showed significantly higher sensitivity toward agent B associating the attribute, *natural* than that with the random agent (t test: $p<.01$). In the comparison based on number of person, 13 participants showed higher scores in the 18 valid results.

Questionnaires

Table 5.20 shows the results of questionnaire investigation. Despite the significant difference in the GNAT test, attentive quiz agent B could not cause positive impression to the participants in questionnaires. Significant difference could not be found in all questions except Q6, "The timings of the character's utterances were appropriate (t test: $p=.09$)" and Q10, "The game progress was smooth (t test: $p<.01$)." Attentive quiz agent B also got negative impressions from the comparisons based on the number of persons, it is less annoying (Q3, 8:13), more passive (Q4, 10:6), less appropriate utterances (Q5, 9:5), and is also worse in Q11, "I would like to response to the character's urges." These results all implied that attentive quiz agent B performed worse in verbal behaviors. This probably comes from the

Table 5.19: The valid GNAT test results of experiment B. “Random” is the random state quiz agent

ID	Random	Agent B	Natural
49	0.128	1.199	A
51	1.060	1.049	R
52	1.407	1.028	R
53	1.060	0.928	R
54	1.561	1.227	R
55	0.456	1.645	A
56	0.967	1.561	A
57	0.524	2.073	A
58	0.511	1.227	A
59	0.910	0.911	A
60	0.800	1.223	A
61	1.290	2.030	A
62	0.757	0.651	R
63	0.379	0.928	A
66	1.036	1.199	A
70	0.911	1.520	A
71	0.896	0.911	A
72	0.639	1.036	A
Average	0.850	1.241	
SD	0.609	0.387	

15 second silent period of attentive quiz agent B. If the quiz is so easy that the participants can answer in short time, the attentive quiz agent B may not have chance to utter. Random state agent does not have this limitation and talked more frequently. About the questions evaluating nonverbal behaviors, the difference was not clear.

Table 5.20: The questionnaire investigation results of experiment B. The questionnaires are evaluated in 7-scale where 1 means smallest degree and 7 means the largest degree. The results are then tested with Wilcoxon signed-rank test. M_R and IQD_R are the median and the inter-quartile deviation (IQD) of the random-state quiz agent. M_B and IQD_B are the median and the IQD of the attentive quiz agent B. "+," "-," and "0" columns mean the number of participants who evaluated attentive quiz agent with higher than, lower than or the same score as the compared random-state quiz agent respectively in each question. "+ rank" and "- rank" shows the mean ranks of positive and negative answers, the larger the number shows higher difference than the opposite answer. "p" shows the two-tailed probabilities of the statistical test

ID	question	M_R	IQD_R	M_B	IQD_B	+	-	0	+ rank	- rank	p
1	The character was friendly.	4.0	1.13	4.0	1.50	8	8	8	6.88	10.12	0.490
2	The character was human-like.	2.0	0.63	2.5	1.00	8	7	9	8.19	7.79	0.750
3	The character's utterances were annoying.	4.5	1.50	4.0	1.63	8	13	3	11.94	10.42	0.483
4	The character was passive.	2.5	1.13	3.5	2.00	10	6	8	9.20	7.33	0.211
5	The character's utterances were appropriate.	5.0	1.00	5.0	1.00	5	9	10	8.90	6.72	0.611
6	The timings of the character's utterances were appropriate.	5.0	0.63	5.0	1.00	5	13	6	9.40	9.54	**0.088**
7	The character behaved in responding to our status.	5.0	1.63	4.5	1.13	8	10	6	9.38	9.60	0.642
8	The character's behaviors were natural.	4.0	1.00	4.0	1.00	9	9	6	7.44	11.56	0.403
9	The character's behaviors were comprehensive.	5.0	1.00	5.0	1.00	6	5	13	6.75	5.10	0.497
10	The game progress was smooth.	5.0	1.00	3.0	1.50	2	12	10	4.00	8.08	**0.005**
11	I would like to response to the character's urges.	5.0	1.00	5.0	1.50	4	10	10	8.50	7.10	0.238
12	The discussion was active.	6.0	0.25	6.0	1.00	4	5	15	4.00	5.80	0.417

Video Analysis

As experiment A-I/II, the video data recorded from the two positions depicted in Figure 5.14 is analyzed as well. Four annotators (three are different to the ones of experiment A-I/II) who are familiar with video annotating but are not involved in the development of this study are asked to annotate the video data. The video data of two groups are selected randomly and are assigned to each annotator (four sessions to every annotator). The video annotation tool, iCorpusStudio was used here, too. The objectives and the algorithms of this study were not included in the instructions for the annotators. The annotators are instructed to annotate the video data as the following conditions:

Participants' attention: as experiment A-I/II, whether the participants paid attention to the agent's verbal utterances and nonverbal behaviors is annotated. The periods during the agent is in all of the five attitude states (*AS*, *AW*, *C*, *IW*, *IS*) are annotated. The labels, *Listen* and *Ignore* are used as how they are defined in experiment A-I/II.

Utterance timings: for the purpose to see whether the agent utters at appropriate timings. When the agent is in *AS* and *IS* states, its utterances are annotated according to the participants' reactions. Note that in *AW*, *IW*, and *C* states, the agent does not make utterances but only performs nonverbal animations. The labels, *Smooth*, *Abrupt*, and *Tardy* are used as how they are defined in experiment A-I/II.

From the observation of the video data, we found that the participants paid nearly *no* attention to the agent (1%) when it only performs nonverbal animations to show its attitude in AW, I, CW states. On the other hand, the participants often pay attention to the agent if it makes utterances, but obvious difference could be found neither between the two kinds of agents nor between the two kinds of attitude states. Table 5.21 shows those results.

Table 5.22 shows how the utterance timings affect the participants' attention in experiment B. The results were similar to the ones in experiment A-I/II. When an utterance is made at a smooth timing, the participants paid attention to it at a probability around 80%, when an utterance is made at an abrupt timing, the participants paid attention to it at a probability around 40%. There was no obvious difference between AS state and IS state, but the utterances made in IS state seem to have less strength in attracting the participants' attention. This is perhaps because the impatient utterances are less pleasing.

Table 5.21: The comparison on participants' attention between attentive quiz agent B and random state quiz agent. The number without remarks represent times

	Random			Attentive		
	AS	IS	AS+IS	AS	IS	AS+IS
Listen	16	9	25	9	14	23
Ignore	8	6	14	4	7	11
Listen (%)	66.7	60.0	64.1	69.2	66.7	67.6

Table 5.22: The influences on participants' attention from different combinations of utterance timings and the agent's attitude state. The number without remarks represent times

	Listen	Ignore	Listen (%)
Smooth-AS	16	2	88.9
Abrupt-AS	10	10	50.0
Smooth-IS	17	4	81.0
Abrupt-IS	6	9	40.0
Smooth	33	6	84.6
Abrupt	16	19	45.7

Summary

The focus of attentive quiz agent B is not in the effectiveness of its utterances but is how it behaves naturally, which is difficult to be observed from the participants' reactions. The results of video analysis coincides this, there was no particularly interesting findings from the participants' reactions. The presentation of the agent merely by nonverbal animations seemed not to be able to improve the feeling of the existence of the agent.

From the objective results of GNAT test and the subjective ones of questionnaires, it suggests a hypothesis that nonverbal behaviors play an essential role in the feeling of human-likeness but they are relatively implicit and do not leave strong subject impression. On the other hand, in the questions related to verbal behaviors, attentive quiz agent B got considerably worse results due to the more conservative utterance policy.

5.5.5 Summary of the Evaluation Experiments

Attentive agent A and B have the same quality of graphics, TTS and non-verbal animations as their compared systems, the only difference was the *timings* to take actions. Attentive quiz agent A and B do actions which are attentive to the participants' status. Still, significant differences could be found in the evaluation experiments. This shows an alternative way to improve the life-likeness of ECAs rather than realistic looking character and animations, by controlling the timings of the behaviors of ECA, positive impressions could be achieved. Second, attentive agent B mainly distinguishes its compared system from the timing of non-verbal behaviors, while agent A distinguishes its compared system with the timings to utter. The considerably better performance of agent B in under-conscious GNAT test may imply that the non-verbal behaviors contribute more to autonomous attitude regarding to life-likeness than verbal ones.

5.6 Conclusions and Future Works

This chapter presented our investigations on the issues involved in the communication with multiple users for ECAs in the context of quiz game. Two approaches are proposed for improving the attentiveness aspect of life-likeness of the quiz agent, a utterance policy and internal attitude adaptive to the users' status. The preliminary evaluation results using GNAT method was encouraging. The ideas proposed in this chapter will then be improved and integrated to next version of our NFRI quiz agent which is deployable in practical exhibitions. The effects of the CLP pointer and how it should collaborate with the CG character are not clear. At present, the following three kinds of settings are possible. We would like to do deeper investigations on their influences to the participants.

1. The CLP pointer has its own personality and behaves as a separate agent.
2. The CLP pointer is an external device controlled by the 2D agent. This relationship should be cognitively recognizable by the participants. For example, by showing an animation that the 2D agent is operating the pointer.
3. The CLP pointer is a part of the 2D agent. In this setting, the appearance and the movement of the pointer need to be carefully designed to prevent the contradiction of the cognition of the participants.

Chapter 6

Visual Knowledge Management System, Gallery

Generally, conventional image/photo managing software features a grid-thumbnail view of the images and relies on the folder structure of the operating system to classify its contents. In our study, we observed that some difficulties are encountered in using this mechanism to deal with large-scale personal photograph collections.

Uncertainty regarding folders. Each photograph can convey several types of information: photographic parameters such as timestamps, focal length, aperture, and shutter speed as well as semantic information such as the location, people involved, or event names. Hence, several viewpoints can be applied to the organization of personal photograph collections. For example, suppose you wish to insert a new photo that was taken during a trip with your friends in the autumn of 2004 into a collection with categorized folders. Which folder would you then place this photo in? Folders named “Trips,” “Friends,” “Autumn,” and “2004” seem to be reasonable choices; however, with the current file management mechanism, you must either select one among them or create redundant copies in the corresponding folders.

Unforeseen changes in organization policies. Let us assume that similar to many other people, you adopt the policy of organizing your photo collection according to the events captured in the photos (Rodden, 1999). In the future, if you wish to find a representatively good photo of one of your friends, it might be difficult for you to recall the event during

which the friend's photo was taken. You will thus need to perform a fair number of linear searches to gather all the photos of this friend, compare them, and then select the best one among them. After this exercise, you may experience the need to create a person-wise category for your collection. Clearly, no single organization policy can be applied at all occasions. Each time a new organization policy is required, extremely laborious efforts will be involved in rearranging the collection to satisfy the new criterion.

Human memory degenerates with time. Often, people cannot clearly recall the actual location of a file last accessed several weeks ago. Moreover, after several months, it might be impossible to clearly remember the contents of a large collection. Generally, people can easily recall recent events; however, with the passage of time, the requirement for assistance in retrieval of older information increases. Unfortunately, most of the current file management mechanisms provide neither appropriate cues that help people recall file locations nor a convenient utility to find a particular file.

Low utility of knowledge assets. Accumulated memories may be valuable knowledge resources during creative activities; however, in the absence of a proper management and reuse methodology, they will be forgotten and will lose their usefulness.

6.1 Gallery System

Gallery is a project aiming to address the problems mentioned in last section. It is intended to support the users in building a sustainable space for externalized personal memories. It features an image content management system that integrates a zoomable overview of images, spatial memory utilization, personal meaningful layouts, and text annotations.

6.1.1 The Design Principles of Gallery

The objective of Gallery is to provide its users a natural environment that functions as an externalized memory space for their mental images, and a mechanism facilitating their flexible knowledge retrieval to store, manage, and reuse their image repository. Instead of simulating the functionalities of the human brain or a model based on human memory theory from the field of cognitive psychology, we adopted an operational approach. This is because the

simulation of the operations of a human brain from the viewpoint of user interface design does not necessarily guarantee an improved performance of a system. Moreover, the actual internal mechanism by which human memory functions is yet to be clearly understood.

On the other hand, we are very interested in the utilization of human spatial memory, the human ability to remember the location of a stored item. We assume that if we can build an environment similar to a personal study room, where an individual remembers the locations of books and stationery, the intuitive environment should be able to enhance the efficiency of content management for long-term use. In some previous studies, it has been proved that the attachment of spatial information to knowledge items makes the leveraging of information retrieval efficiency possible (Czerwinski et al., 1999). These interesting results inspired certain ideas during the design of Gallery.

The following discoveries have been reported in cognitive science research results: people tend to memorize semantic meanings rather than raw text or pictorial information; people display good memory retention particularly in the case of pictorial information, provided they can meaningfully interpret the information; and the memorization of visual information by people improves if the objects in an image interact with each other (Anderson, 2000).

Considering these facts, the following design principles were established for Gallery:

Both image and text information are necessary for representing knowledge. In practice, people can generally visualize an image more clearly from a thumbnail than from a description string (Czerwinski et al., 1999). However, with a single image, various essential semantic meanings such as the time, event purpose, and names of participants cannot be explained. Therefore, we decided to treat an annotated image as the basic unit for knowledge representation; we term this unit a knowledge item. Knowledge items compose concepts, which represent the thought concepts of users; further, they compose the memory space in Gallery.

Using spatial layout as a cue for memory recall. In order to exploit spatial memory, we believe that we should provide a space that allows users to freely place interrelated concepts in an interlinked manner. This meaningful personalized layout should be an effective cue for memory recall and will thus facilitate information retrieval.

The layout of memory storage should maintain temporal coherency. As mentioned previously, the memory storage layout of Gallery serves as a cue when users recall knowledge

item locations. Therefore, it should not be altered dramatically with time. Otherwise, the user will lose her (his) global understanding of the memory space. Hence, despite the availability of many algorithms for generating an automatic layout for knowledge representation, we decided to employ manual layout, which is defined by the users themselves.

Use zoomable user interface for browsing large image collections. According to the suggestions in previous researches (Czerwinski et al., 1999)(Combs & Bederson, 1999), a zoomable 2D user interface can be very successfully applied for browsing large image collections. Therefore, we believe that providing the users a zoomable overview of the memory space comprising thumbnail images can considerably improve information retrieval efficiency.

Visually saving information retrieval steps to enhance user memory. We hypothesized that people recall things by triggering a series of semantic cues ranging from large, unorganized, and abstract concepts to specific and concise knowledge fragments. Moreover, many people tend to organize things by classifying them in hierarchical categories. Further, considering the simplicity of the implementation of a preliminary prototype system for the purpose of evaluating its feasibility, we decided to use the tree representation of knowledge space in the current prototype system.

Link interrelated images as stories. Sharing among friends is a common use of photos, and people like to describe events through a slideshow of photos (Rodden, 1999). Since digital photos can be copied at no costs, people are more willing to do that by using digital photos (Rodden & Wood, 2003). We believe that linking interrelated pictures as an ordered group, which we term a story, is richer in expression than fractions of information in the form of pieces of pictures. Thus, they should serve as an effective medium for knowledge exchange between users.

6.1.2 The User Interface and Operations of Gallery

In Gallery, the memory space is depicted on a 2D interface. The left corner of the screen is called the importing area, which serves as a working area for the user to import new images to Gallery. The main component of the display is the memory space in Gallery; it is the central component where users browse and manage their image repositories. The

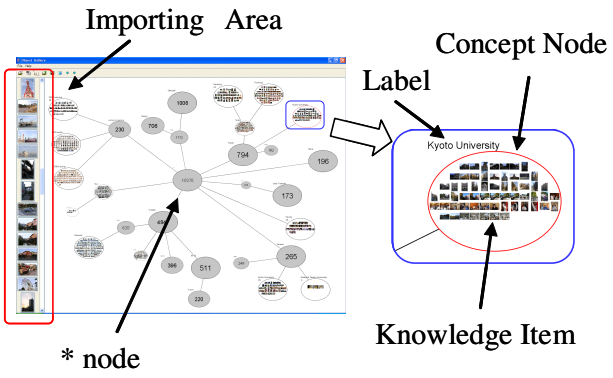

Figure 6.1: The user interface of Gallery

Gallery memory space comprises concept nodes that represent the user's thoughts. A concept node that corresponds to a single thought is drawn as an ellipse in the memory space area. Knowledge items are displayed as thumbnails within a node. An overview of the Gallery user interface is shown in Figure 6.1.

At first use, Gallery presents the user an empty universal concept node labeled as "*." The user then imports new images to the memory space and assigns optional keywords to the images in a dialog box. These newly added images will then appear in the * node as thumbnails, and they are laid out to occupy the node containing them in the largest possible size and in row-first chronological order by default. These thumbnails can then be moved to arbitrary locations by drag-and-drop mouse operations.

The user creates a new concept node by dragging the mouse pointer from an arbitrary node and entering a string filter in a dialog box presented after the location of the new node is determined. The filtering string then becomes the label of the new concept node. Gallery uses this filtering string to match keywords, annotations, the file path, and the last modification date of the knowledge items in the parent node. Items coinciding with the filtering string will then become the contents of the newly created node.

A filtering string can consist of multiple keywords and temporal or negative propositions. Filtering strings are evaluated as the logical AND result of each component term. A negative proposition is prefixed with "!," while temporal propositions are specified by reserved keywords "Y:," "M:," "D:," and "W:." For example, a filtering string, "Y:2004 M:12 D:25 W:SAT" represents Christmas Day of 2004. When images are imported to a

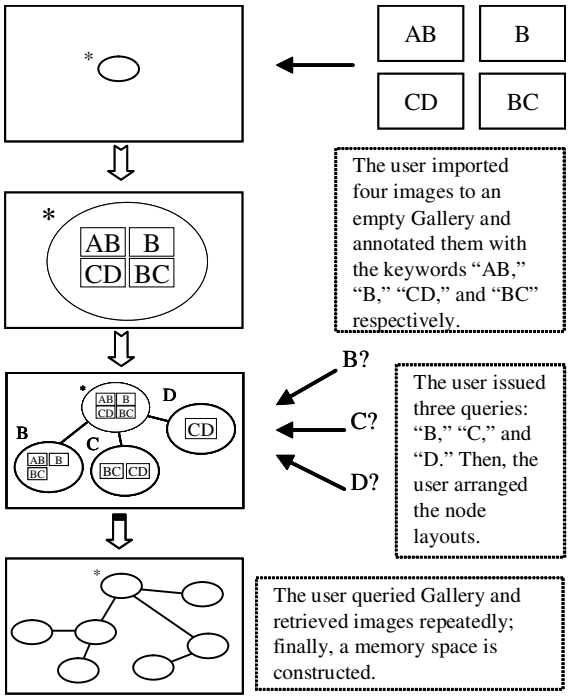

Figure 6.2: The process for constructing a memory space in Gallery

non-empty memory structure, new images are individually verified based on the filtering strings of each node beginning from the * node and moving toward the matched nodes.

Since the user retrieves information from the memory space and places newly generated nodes repeatedly, a tree structure will finally be constructed. The process for constructing a memory space from an empty * node is shown in Figure 6.2. This example depicts a case where a user imports four pictures, annotates them with the keywords "AB," "B," "CD," and "BC," and then queries Gallery with the keywords "B," "C," and "D."

This 2D image space can be smoothly zoomed and browsed, and every node and knowledge item can be freely placed by drag-and-drop operations. Concept nodes other than the * node can be deleted manually if they are considered redundant by the user. In addition, we added several widgets to improve the browsability of the memory space such as double clicks move concept nodes to the center of memory space or trigger a detailed view of an image. A triple click positions the view so as to center the node and then zooms in to enlarge the node to as large a size as possible such that its contents are clearly visible. Concept nodes that contain numerous knowledge items such that the thumbnail sizes are smaller

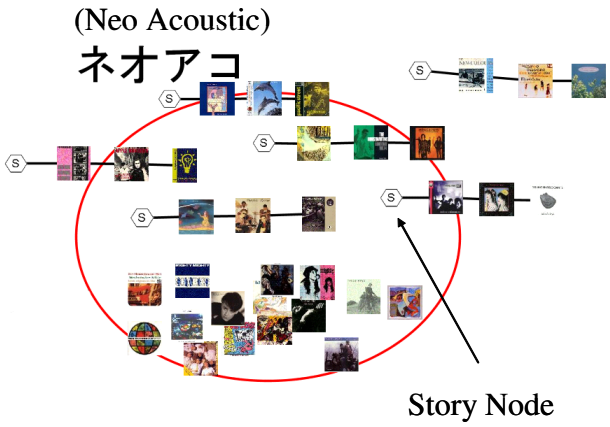

Figure 6.3: Stories in a concept node

than an identifiable threshold are colored in gray, and a numeral indicating the number of contents in that node is displayed. In order to provide space for more frequently accessed nodes, nodes that are less frequently accessed gradually shrink in size. We decided to enable the nodes to change their sizes automatically rather than through user-defined operations, because explicit operations can be considered tedious when the number of nodes is large. Moreover, this situation mimics the behavior of human memory, i.e., people gradually forget things if memories are not recollected over long periods; they do not choose a memory to be forgotten.

All operations and movements of objects in Gallery are shown in animation on the same screen surface so that the user is aware of the activities while having an integrated feel of the system. Further, the disturbances caused by window operations are eliminated.

In addition to the basic image-library-browsing functionality, stories can be created by single-click mouse operations. The user can then annotate each image with a brief description of the story captured in the photo. Stories can be used as units for slideshows and can be imported to or exported from Gallery. As an example, a node containing linked stories is shown in Figure 6.3. The user has defined six stories inside the node “Neo Acoustic.” Note that the stories are preceded by dedicated story nodes, and interrelated images are linked in a specific order by a single line. A single image can be linked to multiple stories, and the story nodes serve as the identifiers of the headings of stories. When the user double-clicks on a story node, a slide show is triggered in an external window. The story annotations

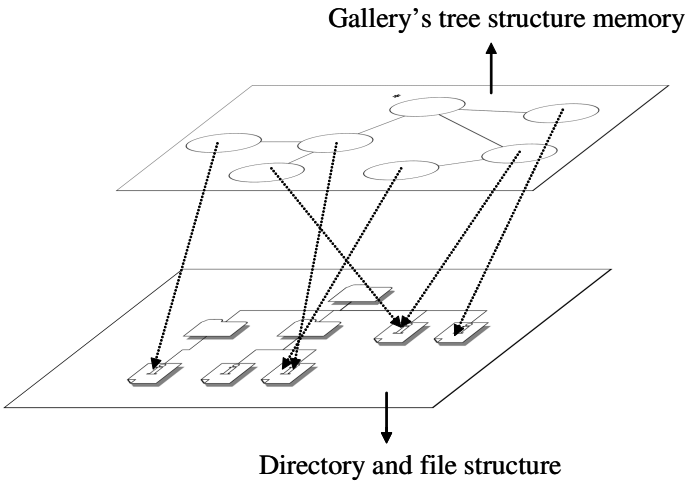

Figure 6.4: The memory space structure of Gallery

describing a photograph frame can be read and edited in this window.

We also developed a dedicated simplified Web server for Gallery. The user simply places her (his) memory space configuration files in a specific directory on the Gallery server and launches the server. These memories are then published on the Web where they can be viewed by the user's friends from remote computers.

6.1.3 The Fundamentals of Gallery

Knowledge items in Gallery function as soft links in the UNIX file systems: an actual file body in the OS file hierarchy can have multiple instances in the Gallery memory space. This concept is shown in Figure 6.4 where a single image is linked by multiple knowledge items. The concept not only serves to save disk space but also reduces redundancy by the method of "edit once and all are affected" for keywords and annotations. Since multiple keywords can be associated with each knowledge item, overlapped categories of different concepts can easily exist simultaneously. Therefore, the user will not hesitate in determining the folder in which a new image must be placed, and s(he) will construct concept nodes according to her (his) choice in case the target image cannot be ascertained from the existing concept nodes. Another benefit of this structure is that a new category of images or a new subtree of the * node can be built in seconds. Therefore, the alteration in the hierarchy of the existing

content or the addition of unforeseen changes in the organization policy can be carried out effortlessly.

Based on the manner in which people think, as described in section 6.1.1, we expected that when a user tries to retrieve a knowledge item, s(he) first recalls the most likely keywords that may refer to the image item and then browses through and inspects the results of this trial search. If the resulting child node contains a considerable number of hits, such that the user cannot find the desired image, s(he) may refine her (his) search and try again until the desired information is obtained. We say that the path from the * node to the target knowledge item records the user's thoughts during the recall process.

The memory space will gradually expand. Moreover, with the user's thinking process, the grouping of photos, annotations, placement of concepts, and operations on photos augment the memory space, and it finally evolves into an externalized version of the mental image of the user's memories as captured by the photos. We expected that further information retrieval can be performed more easily and efficiently because this personally meaningful layout arranged by the user herself (himself) will function as a useful cue for memory recall. We believe that with this improvement in the information retrieval efficiency, the utilization of past knowledge also improves. We expect that the discovery of forgotten memories will be very interesting and may also stimulate creative activities.

Finally, it is considered that spatial representation approaches for knowledge storage will begin to lose their advantages when the collection size increases. In Gallery, to resolve this situation, in addition to the spatial layout, we provided text labels for node filters, thumbnail sizes, node sizes, and number of items. We expect that these labels will function as cues for recalling the node contents during the image retrieval performed by the user.

6.1.4 Implementation

In the implementation phase of Gallery, we aimed at the management of large personal image repositories. For this purpose, the smooth browsing and processing of large image collections, which at least correspond to a scale of several thousand images, were the basic requirements that we wanted to meet. At the same time, since the target users were individuals requiring the system for personal use, high-end hardware environments should not be a requirement. Therefore, the most difficult problem we faced was in striking a balance

between the usage of memory and a reasonable performance while accommodating as many images as possible.

During the implementation of Gallery, exclusive efforts went into memory management and determination of parameter values. Since the target application of Gallery is the management of personal image collections, the photographer is usually the user or a person acquainted with the user; therefore, the user should be familiar with these photographs. We found that in such cases, people are able to identify even the thumbnails that have a relatively small size based on the color distribution cue of each thumbnail or the thumbnail set of a node. In the preliminary version of Gallery, we chose to apply a fixed-size threshold to simplify the process and maintain a common standard in our evaluation experiments. Based on empirical results obtained for relatively young subjects from our laboratory, we found that a threshold comprising a 6×6 pixel square is adequate. However, if this value is adjustable, it would be more flexible for a wide range of user ages. When the thumbnail image size is below this threshold, the user is no longer able to identify a photo even if s(he) had taken the photo herself (himself). Therefore, we cut off the detailed display of thumbnails of a node below this threshold to reduce the computations of the unidentifiable thumbnails. The nodes with thumbnail sizes smaller than this threshold are filled with gray color and labeled with a numeral indicating the current number of knowledge items present in them in order to make them more recognizable. For the same reason, the largest thumbnail size of a knowledge item is restricted to a 72×72 pixel square.

The results of these efforts were impressive and satisfying, as observed in the preliminary test of the Gallery prototype: we imported a personal photo collection of one of the authors, which contains 11,454 pictures. After organizing these pictures, the author created 30 nodes and 39,071 knowledge items. Despite the fairly large image collection, Gallery functions effectively and the photos can be browsed very smoothly on a 1.2-GHz Intel Pentium III laptop computer, which is not very fast, and less than 256 MB of memory was occupied on the Java virtual machine. Based on this result, we believe that Gallery should at least be capable of dealing with image repositories of the scale of several dozens of thousands of pictures on normal desktop machines. At present, this corpus is the largest one available to us. The * node of the memory space created by the author is shown in Figure 6.5.

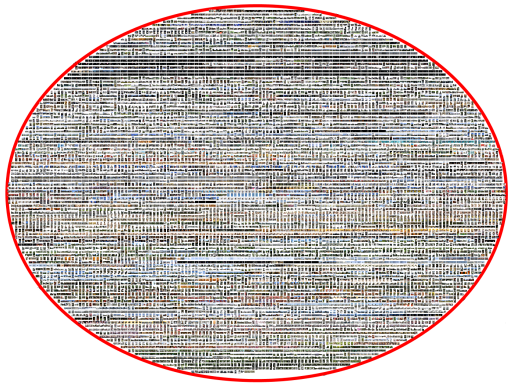

Figure 6.5: The * node of the 11,454 photo corpus; the thumbnails are very small, but identifiable on a 1600 $\times$ 1200 monitor, and are larger than our threshold of a 6 $\times$ 6 pixel square

Moreover, in order to make it possible to browse the Gallery memory space using normal Web browsers, we implemented the Gallery prototype system in Java. The prototype system can work as a stand-alone program on a local machine or as a Java applet embedded in a Web page, and thus, can be browsed from a remote machine. Currently, many users utilize a network environment guarded by a firewall that bans connections through unknown ports for the purpose of security. Considering this factor, we decided to use HTTP as the communication protocol between the Gallery applet and the server where the actual memory space data is stored. Initially, we developed a prototype system using a commercial Web server, Apache (Apache Software Fnd., 2004), a database system, MySQL (Sun Microsystems, 2004b); and J2EE (Sun Microsystems, 2004a). Although this prototype system functioned effectively, we soon realized that it is very difficult for a normal user who is not computer savvy to set up such an environment. Therefore, we developed a simplified Web server dedicated to the Gallery applet. It returns the thumbnails and memory space structures as HTTP responses to requests from the Gallery applet.

Concept nodes change their sizes according to the following linear function:

$$\frac{(MAX - unvisited) \times (RATIO - 1) + MAX}{MAX \times RATIO} \tag{6.1}$$

MAX is a constant that scales the rate of change of node sizes and is fixed at an adequate value of 2000 in this case, based on empirical results. unvisited is a variable associated with

each concept node, which records the number of times the node is untouched while the user clicks on the other nodes. The maximum value of unvisited is limited to the value of MAX so that the linear function does not produce a negative result. RATIO is a constant that controls the size ratio between the largest node and the smallest one and is set to 3. As a result, concept nodes will linearly shrink to one-third of their original size based on how infrequently they are accessed by the user. Every time they are accessed, they are restored to their maximum size. As another option, MAX can be set as a variable that increments its value when the user clicks on any node; however, this method has a drawback in that the sizes of all nodes and thumbnails need to be recomputed each time the user clicks on a node. The performance of the system will deteriorate to unacceptable levels when the number of image contents increases. Moreover, since MAX will exceed the representable range after long-term use, we decided to use a constant in the node shrinking function. For the convenience in possible knowledge exchange between Gallery users, human readability, and for platform independence, the tree structure of Gallery memory space, keywords of images defined by the user, and annotations are stored in an original XML format. Finally, to reduce the complexity of implementation, the zoomable user interface of Gallery is developed based on the Piccolo (Bederson et al., 2004) library, which was developed at the Human-Computer Interaction Lab of Maryland University.

6.2 Evaluations

In addition to the in-house test of Gallery described in the previous section, we conducted two evaluation experiments in order to understand its real-world feasibility.

6.2.1 Monitor Study

First, we distributed the Gallery prototype to six monitors, five males and one female. These monitors arrived from three countries, Japan, Taiwan, and China, and belonged to two institutions, Kyoto University and Taiwan University. Their ages ranged from 22 to 33 with an average age of 26.8. All of them had a background in computer science. They were asked to use Gallery to organize their personal photo collection and were encouraged to create

Table 6.1: Constructed Gallery memory spaces in the monitor study (AVG. denotes average, S.D. denotes standard deviation)

	User A	User B	User C	User D	User E	User F	Average	S.D
Number of Photos	101	769	259	481	428	511	424.8	208.7
Number of Nodes	7	44	12	12	25	35	22.5	13.5
Links per Photo	2.52	2.62	2.56	1.99	5.11	7.95	3.8	2.11
Keyword Variation	166	46	15	40	40	39	57.7	49.4
Keywords per Photo	4.39	1.16	2.52	1.19	3.45	6.08	3.13	1.75

stories by using text annotations. Two weeks later, we gathered the data on the memory space created by the monitors. A questionnaire was used to interview the subjects; the questionnaire attempted to gauge their level of satisfaction and impression while using Gallery in these two weeks.

Table 6.1 shows an overview of the memory spaces constructed by the six subjects. We found that the individual differences between these subjects with regard to photo management were obvious. We consider that this is a result of the very flexible content management offered by Gallery. Two significant facts can be ascertained from this table: the average number of soft links per photo is fairly high at 3.8, and on an average, 3.13 keywords are assigned to a single photo. This result implies that in this experiment, on an average, one photo appears in 3.8 concept nodes. Despite excluding a possible redundancy in the root node, it is still implied that a photo usually has several overlapped semantic properties, and the number of redundant duplicates required may be the same as that in single-hierarchy file management architectures such as conventional image managers.

The most important feature of Gallery that differentiates it from the other image managers is that it allows a personally meaningful layout of images. Therefore, in this experiment, we are primarily interested in determining the manner in which the users arrange their concept nodes. We observed that most of the monitors simply placed new nodes in radial directions from the * node. We think that this is a side effect of the provision of the * node. The existence of the * node provides two functions: the * node and the edges radiating from it help to ensure that all the nodes can be easily found in the space, and it serves as a container for unused contents. However, since the * node may sometimes become an obstacle

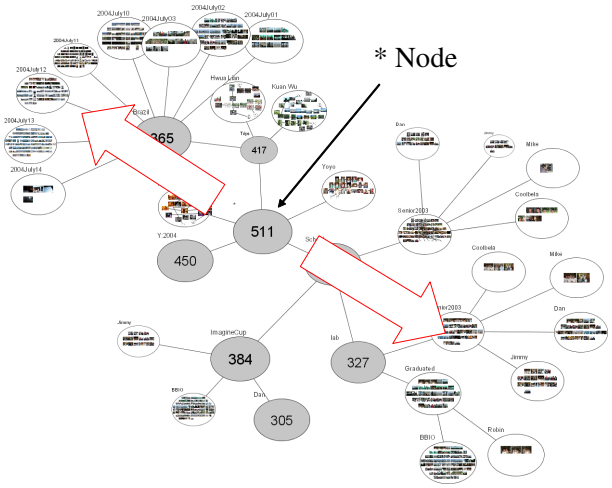

Figure 6.6: A radial layout created by user F, a typical layout of Gallery users

in node layout, a further reconsideration of this factor may be necessary in the next version of Gallery. A typical example of such layouts is shown in Figure 6.6, where the memory space created by monitor F is depicted.

In contrast, the other users learned to utilize the customizable layout more effectively. For example, user B dragged the nodes away from the * node and provided more space to accommodate the layout of the other nodes. Moreover, he organized his photo collections relative to their actual geographical locations. His memory space is shown in Figure 6.7. The concept nodes clustered photos taken in Japan, U.K., Italy, and other places into four groups. Within each group, the nodes were further clustered according to the relative positions of the cities.

Although Gallery is designed for the management of personal photo repositories, user A used it to sort the CD jacket graphics of his collection. His memory space is shown in Figure 6.8, where the CDs are sorted according to their categories and publishing year. We did not originally envision this type of application during the design phase. However, this example shows us the possibility of using Gallery for managing real objects by importing their photos. It also shows that the system can be used for other applications as well.

After the two-week evaluation period, we interviewed the six monitors with a questionnaire comprising selective and descriptive questions. The first part of the questionnaire

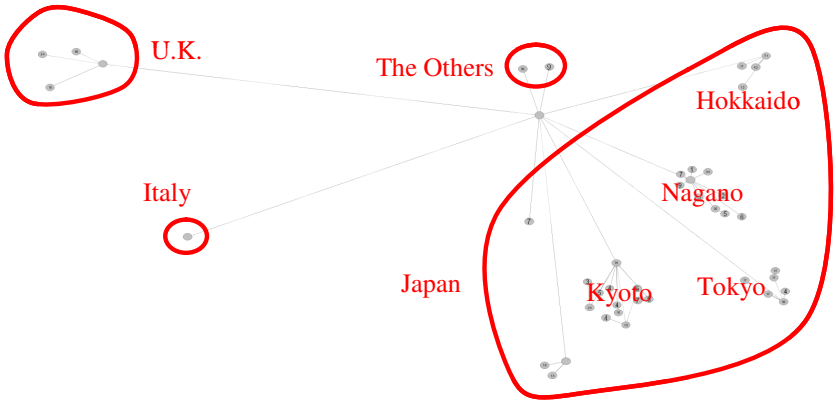

Figure 6.7: The memory space created by user B. Nodes are placed according to the real world locations of their contents

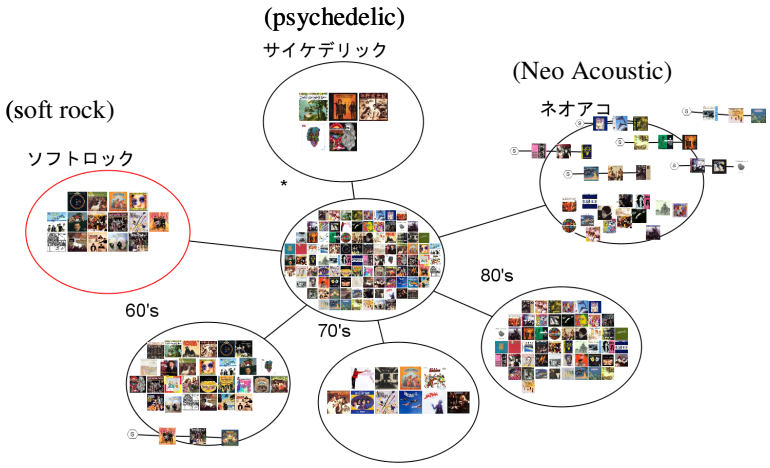

Figure 6.8: The memory space created by user A. CD jacket pictures are imported instead of photos

Table 6.2: Result of 5-point-scale questions to monitors about their satisfaction with Gallery (1 = "strongly disagree," 2 = "disagree," 3 = "I don't know," 4 = "agree," 5 = "strongly agree")

ID	Question	AVG.	S.D.
Q1	I like Gallery	4.2	0.7
Q2	Gallery is a useful software	3.7	0.7
Q3	The Gallery user interface is intuitive	2.8	0.7
Q4	Photos can be efficiently sorted using Gallery	3.3	0.8
Q5	It is easy to sort photos using Gallery	3.8	0.7
Q6	It is easy to learn how to use Gallery	3.5	0.8
Q7	It is easy to locate a particular photo using Gallery	4.0	0.6
Q8	It is easy to remember photo locations using Gallery	4.0	0.6
Q9	Browsing photos with zooming user interface is useful	4.8	0.4
Q10	Keyword search is useful	4.5	0.5
Q11	Using node size to distinguish between the access frequencies of nodes is useful	2.8	0.4
Q12	The story feature is interesting and I intend to use it frequently in the future	3.7	0.7
Q13	The overall thumbnail view is useful in finding a particular photo	4.2	0.4
Q14	Arranging the layout of the nodes by myself helped me remember the photo locations better	4.2	0.9
Q15	I feel my photo collection is better organized than before	4.5	0.5

comprises a series of 5-point-scale questions and the results are listed in Table 6.2. Questions Q1 to Q8 ask the subjects about their overall impression while using Gallery and questions Q9 to Q15 investigate the manner in which its individual features contribute to its effectiveness. From these results, it is clear that the basic concept of Gallery, including the integration of keyword search (Q10, 4.5) and zoomable user interface (Q9, 4.8) for photo collection browsing, worked effectively. The monitors stated that by arranging the layout by themselves, they were able to remember the photo locations more easily (Q14, 4.2), and they were able to locate the target photos easily (Q7, 4.0). In addition, all the monitors agreed that after using Gallery, their photo collections are better organized than before (Q15, 4.5). Only two questions scored below 3.0. The first question was about the effectiveness of the shrinking size of infrequently accessed nodes, some monitors said that they did not notice the changes in the node sizes. We think that this is due to the short evaluation period of two weeks, where the change is barely noticeable. Further, the appropriate parameter values for the size changing function explained in section 6.1.4 might have to be reconsidered. The second question was about the intuitiveness of the Gallery user interface, some monitors said that the zooming user interface in Gallery is not so Microsoft Windows like, which they were already accustomed to; therefore, they found initially, it was not easy to use Gallery. The others said that after familiarizing themselves with Gallery in a short time, they faced no difficulties and it was very easy to use.

The second part of the questionnaire comprised a detailed interview. We obtained the following findings from this part. Five of the six monitors said that before using Gallery, they indeed encountered difficulties in managing photo collections using conventional image managers such as Explorer, which is inbuilt in Microsoft Windows XP. They indicated that the difficulties include easily forgotten file paths, locations of newly added files that were hard to determine, and the lack of an overview of all the images at the same time. Even encountered such dissatisfactions, five monitors answered that they just continue to use Explorer without attempting to find a more capable image management utility.

Although none of the six monitors published their memory space on the Web, we asked them about their interest with regard to the Web publishing feature. Four of them expressed a desire to share their photos with their friends and only two expressed a desire to publish their photos publicly. This result was interesting and it provided us a design direction for

the software. It indicated that people like to share memories; however, in most cases they only wish to share photographs with their friends due to privacy considerations.

Further, the response of the users with regard to the story feature was diverging. Some monitors said they did not understand the usefulness of stories. One monitor used it only for creating slideshows. An interesting finding is that two monitors said they did not find creating stories for themselves very interesting. This may be because the photos in the collection were still fresh, and therefore, they were self-explanatory. Hence, there was no need to annotate them. On the other hand, they enjoyed browsing through other users ' stories and could quickly derive the overall image of an unfamiliar photograph collection.

Finally, four of the monitors said that although keyword annotations helped immensely in subsequent content retrieval and memory recall, it was still laborious to assign keywords manually. They said that even Gallery should provide a grouping keyword assignment feature, and some level of automatic keyword assignment was still desirable. Automatic keyword assignment is beyond the initial research scope of Gallery and is currently left as an open problem. However, we intend to incorporate a flexible automatic or semiautomatic annotating mechanism that uses information sources such as e-mails (Lieberman et al., 2001).

6.2.2 Effectiveness Evaluation

The objective of Gallery does not include pursuing performance during speedy browsing. However, we conducted an experiment to understand its contribution to memory recall in comparison to that in a conventional photo manager. The performance of ACDSee 7.0, which is a representative commercial digital image manager and is considered to be a very popular and fast image browser in the market, was selected as the baseline. ACDSee features a traditional photo manager with a very fast grid thumbnail preview, keyword assignment/search, and a calendar view.

There are two sessions in this experiment. The subjects are asked to select ten of their favorite photos from a corpus in the first session. In the second session held one week later, they are asked to find these photos. The time the subjects spent in searching photos in the second session was measured for comparing the two software. In order to unify the experiment conditions, we prepared a common photo corpus consisting of 600 photos that were annotated with 69 different keywords (an average of 3.4 keywords per photo). These

photos did not belong to a specific genre and included landscape, building, people, and event photos.

Eight male and two female subjects participated in this experiment. Their ages ranged from 22 to 24. None of these subjects exhibited any obvious disorders in image recognition, mouse operation, and memory. All of them were senior students of computer science related departments in Kyoto University and were experienced computer software users. After a brief introduction to the usage of these two software, the ten subjects were divided into two groups by drawing lots and asked to browse and select ten of their favorite photos from the 600-photo corpus in 20 min. A list of all the keywords was provided to all the subjects for reference. Members in one group used Gallery and those in the other group used ACDSee.

We divided the subjects into two groups without asking them to test both software because we wanted to prevent the users from discerning the intention of this experiment, which could affect the experiment results. Moreover, these subjects did not have any experience with the software tested, and they had not viewed the photos earlier. All subjects used the same desktop PC so that the differences in environmental factors could be neglected. This PC was equipped with a 20-in LCD monitor, and its display resolution was set to 1600×1200. ACDSee was configured to display 30 thumbnails at a time. None of the subjects were involved during the development of Gallery. In the first session, the subjects were unaware of the basis for the experiment and the procedure to be followed in the second session. Therefore, it was considered that the subjects did not intentionally memorize the photos selected by them.

One week later, the subjects returned to our lab and were asked to use the same PC to find the ten photos that they had selected one week ago. In the preliminary experiment of Gallery, although the photos in the test corpus were not directly related to the subject, one subject scored a fairly high recall rate of 50% without requiring any hints other than the Gallery memory space saved one week ago. In contrast, most of the other subjects reported that they could not obtain a strong impression from unfamiliar images in a short time and were unable to figure out all the photos correctly. By using ACDSee, the recall rate is supposed to be further low. Therefore, to prevent an unlimited time for memory recall, in the second session, all the subjects were shown a thumbnail printout of the pictures they had selected previously. In addition, in order to exclude the factors arising from the absolute

advantage of Gallery keyword labels, the ACDSee group subjects were provided a list of keywords that they had used previously.

The results of this experiment were as follows. The ACDSee group spent an average of 247.3 seconds, while the Gallery group spent 197.6 seconds; therefore, the Gallery group was faster by more than 20%. Both software provide similar features including thumbnail view, keyword annotation/search, and calendar search of pictures; however, Gallery exclusively differs from ACDSee in that it continually records the user content retrieval history visually. In contrast, conventional image managers such as ACDSee do not provide any cues after image retrieval. Therefore, Gallery users can recall file locations rather easily without having to scroll through thumbnails to try and find the required information. Moreover, note that Gallery was designed for managing the user's personal photo collection, where the photographer is the user herself (himself). Thus, the user will have a stronger impression of her (his) photos in her (his) memory. Based on this argument, if the subjects could locate target pictures considerably faster from an unfamiliar picture collection, they should be able to perform even better with their own picture repositories. Furthermore, the test corpus was not a very large collection, and we expect that the difference in performance would be more obvious in a larger image collection.

6.3 Conclusions and Future Works

This section described Gallery system that is intended to support the management of personal image repositories that store the valuable memories of a user. We introduced the Gallery system and discussed the results of two evaluation experiments related to the feasibility of Gallery. One was a monitor test and the other was a comparison with a conventional image manager. The results showed that the basic design concept of Gallery with a feature that allows the users to personalize photo collections in a semantically meaningful layout on a zoomable surface considerably improved the efficiency of image information retrieval. These two experiments were performed with a relatively small number of subjects and the experiment periods were relatively short. However, they provided some hints and new directions for the development of Gallery. It is also possible that the results will promote researches in the management of large personal image repositories.

Many image managing applications have been proposed previously and neither keyword searching nor zooming viewer of image collections is a new idea. In comparison to previous studies, Gallery shows its advantages and effectiveness with regard to the following aspects. It retains the search steps and visually depicts them in the memory space; in our experiment, this method was proved to help users remember their personal collection better and facilitate further image retrieval. Image browsing and viewing functions in Gallery not only facilitate smooth image browsing of folders but also allow the users to organize their image collection in a semantically meaningful layout. This makes subsequent image searching much easier due to the spatial memory of human beings. Both these conclusions were proved in the experiments described in section 6.2.

In addition to some small defects found during the process of the evaluation experiments, we intend to consider the following improvements in the future.

First, the tree structure used in Gallery sometimes limits the freedom of user layout. We are considering redesigning the memory structure of Gallery to a more expressive, flexible, and intuitive representation structure in the next version. Second, although manual layout leads to better memory retention, if the users import a large number of images at one time, it will be laborious to arrange the layout of all the contents. Therefore, some level of automatic layout algorithms that do not generate arbitrary layouts and some handy widgets supporting content manipulation in accordance with the user's hand gestures are desirable. Third, it can be easily considered that many concept nodes will be generated in long-term use. Although redundant nodes can be deleted by the user, this cannot be considered as the real solution to this problem. Therefore, we intend to develop further abstractions such as islands of concept nodes. Fourth, the use of the current system is limited to digital photos with text annotations functioning as knowledge representation media; we intend to expand the system to accommodate a wider range of knowledge media such as video clips, sounds, word processor documents, presentation slides, and e-mails. Lastly, inserting keywords manually can be considered laborious and tedious; in the future, we intend to develop automatic keyword generating mechanisms.

Chapter 7

Discussions and Future Works

This chapter ends with a proposal of the Circulating Knowledge with Virtual Agents (CINOVA) framework that aims to facilitate the knowledge circulation process between institutions and their public audiences.

7.1 The Relationship between GECA and SAIBA

The works of SAIBA framework that we described in section 2.1.2 has a strong relationship with this study because it shares the same goal partially, i.e. to provide common standards in developing ECAs. GECA distinguishes SAIBA from its larger scope in building a whole ECA system, i.e. from input to output rather than the output-only one of SAIBA. GECA also distinguishes from SAIBA from the attempt to provide a complete solution for building ECAs including the integration middleware, utility libraries, and character animation player where SAIBA emphasizes in description language designs. The ideas of SAIBA, especially the ones included in BML are not necessarily superior than previous languages in all aspects, but its initiative was strong and finally can be expected to become a de facto standard in the future. We are watching the progress of SAIBA framework, especially the still unmatured FML activity. We would like to contribute our ideas to it.

GECA and SAIBA are not competitive to each other. Since SAIBA is only an idea of a topology design and language specification, it can actually be implemented on top of GECA platform (see section 3.4). Current character animation player only accepts GECA's

own format of animation description language that is similar to calling animation functions with parameters. They can actually be described with *macros* of BML descriptions. A GECA-BML compiler is possible to be implemented for this purpose.

7.2 The CINOVA Framework

Institutions demand an effective way to disseminate their knowledge and information to public audiences. City malls want the citizens to understand the regulations of how to dispose large garbages or how to state yearly income and calculate the tax, the Ministry of Health wants people to notice the spreading infectious diseases and know how to prevent it, the Meteorological Agency wants people to pay attention to a coming typhoon or understand the mechanism of earthquakes, a science museum wants its visitors to understand and experience the principles of mechanics, research institutes want to introduce their results and make difficult theories easily understandable to the public. At the same time, institutes want to get the feedbacks from the public, what people want to know and what was not clearly conveyed. In a large institution, usually there are many experts who possess specific aspects of knowledge but do not know the others well. The scattered institution knowledge has to be stored, well managed and organized to be useful and can be reused to create new values (Alavi & Leidner, 1999). Two essential issues emerged in the knowledge circulation, the first one is how to efficiently store, organize and reuse large amount of knowledge that is scattered among many experts, the second one is how to efficiently disseminate information to and get feedback from public audiences.

This section presents the Circulating Knowledge with Virtual Agents (CINOVA) framework that proposes the integration of visualized knowledge management systems (VKMS) and life-like virtual agents for these two issues. The basic requirements of a knowledge circulation framework is the storage and a common presentation of knowledge. The knowledge representation should be able to describe various principles of knowledge and can be easily accessed by many experts who work on different computer systems and have different preferences on user interfaces. The core of CINOVA framework is a back-end knowledge repository of the whole institution and is shared by all of the experts (Figure 7.1). The basic unit of the common knowledge representation stored in the knowledge base and exchanged

among the subsystems is so called *knowledge cards*. As proposed in (Kawakita, 1975), describing pieces of knowledge into card media is an efficient way for one or a group to organize known information and to create new thoughts. A knowledge card in CINOVA is a metaphor of such a card that represents a piece of knowledge and is composed with a fragment of XML text and one image. It is simple but is a general representation of knowledge in any principle and can be processed by various applications on various operating systems. it can be a research node, an e-mail, an introduction of an insect, an experience of a person, a quiz about animals, an introduction of a sightseeing spot and so on. Multiple relevant cards can be further linked sequentially to be a *story* to form a presentation of specific topic.

It serves as the knowledge repository of the whole institution. Its knowledge contents inside it are contributed from the institution members. When the amount of the knowledge contents gets large, they become difficult to be handled and be thoroughly understood. Therefore, information visualization techniques are applied to provide efficient interfaces for the operations like uploading, organizing and authoring of the knowledge repository that may contain many thousands of knowledge cards. Several such visualized knowledge management systems (VKMS) can be connected to the same shared repository and provide different abstract views for the experts' convenience.

These knowledge contents are then presented by life-like virtual agents as the interface toward end public audiences. Life agents are considered particularly effective and intuitive for non-expert public users because no extra training is required and allow people to use daily-life communication skills to interact with them. Two ways of presentations are anticipated, the presentation on the Web which is more limited in functionalities but has broader audiences, on-site presentation in exhibitions which is more interactive and allows the visitors to directly try and experience so that deeper understanding can be expected. There are four systems already developed for different needs and presented in following sections. It is connected by four subsystems that are for different purposes and are presented in following sections. Two visualized knowledge management and contents authoring systems (VKMS), the Gallery system described in chapter 6 and a 3D Sustainable Knowledge Globe (SKG) (Kubota et al., 2007). One Web based avatar presentation system EgoChat (Kubota et al., 2004) and GECA agents. A story is the basic unit of a presentation. There are four user classes in CINOVA framework.

Knowledge contents providers. They are the experts in the institution who possess specific knowledge in their minds and are willing to contribute it to the others in the institute or to disseminate it to the public. For example, in the case of NFRI, they are the researchers of food science. One provider may describe a piece of knowledge as a knowledge card and upload it to the shared knowledge base by using one of the VKMSs.

Presentation contents creators. They are the people who belong to the institution and create agent presentation contents (stories) by authoring the knowledge cards stored in the shared knowledge base by using one of the VKMSs. Depending on the target presentation agent system, the knowledge of how to compose expressive and natural non-verbal behaviors of the agent is required, they may be or may not be the knowledge contents providers.

Grouped exhibition visitors. They are the users who actually visited the exhibitions of the institution or the museum. From our observations in NFRI, the visitors go to exhibitions are usually in groups like students in the same class, friends, couples or families. In the CINOVA framework, we meant to provide these visitors immersive and multi-modal interactions with the knowledge presenting virtual agents. The setting of sensor devices, microphones or cameras that capture the activities of the visitors and 3D graphics that required high-end machine are possible.

Individual Web visitors. They are the people who access the Web site of the institution remotely. In the Web environment, the setting of sensor devices and the timing control of the agent's behaviors are not practical and thus the agent's functionalities are more suppressed.

These users exchange, share and acquire knowledge via knowledge card media through the CINOVA framework. The experts provide their knowledge to the knowledge base, the creators author the cards to presentation contents (stories), the knowledge is then presented by virtual agent systems instead of the staff of the institution. The knowledge consumers (visitors on-site or from remote) acquire their demand knowledge via the interactions with the virtual agents who are never tired and can serve queries in all aspects as long as the answers can be found in the knowledge repository rather than a human exhibitor who is usually only an expert of certain area. The visitors can listen to the presentations done by the virtual agents, ask questions if they do not understand, or play quiz games with a virtual agent. If the agent can not answer a query issued by the user, that question can be sent back so that the knowledge providers and creators can produce new contents to answer it

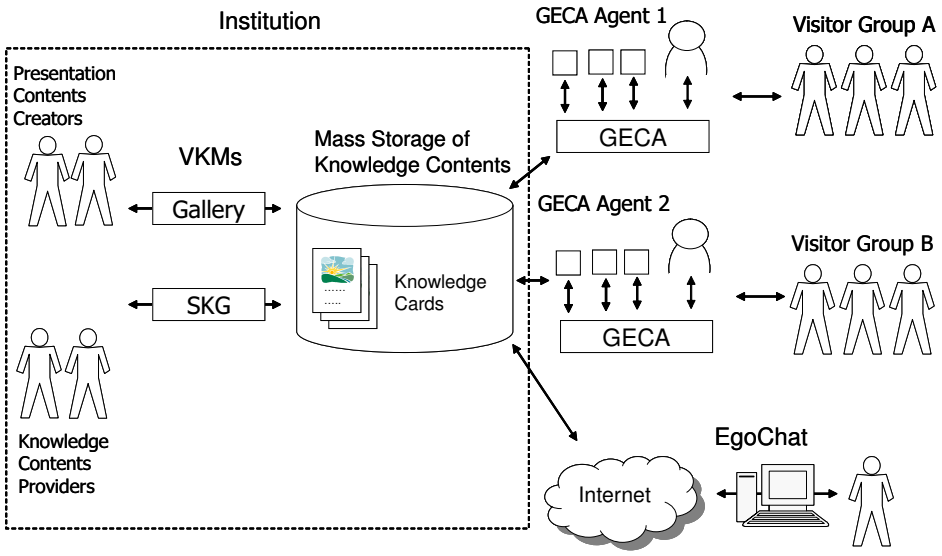

Figure 7.1: The concept diagram of the CINOVA framework

for queries in the future. This forms a circulation of knowledge and is considered to be able to facilitate the communication between the institution and the public audience. The knowledge is made actionable and can also facilitate the institution to create new knowledge.

Chapter 8

Conclusions

This dissertation proposes the Generic Embodied Conversational Agent (GECA) framework that is a general purpose development framework for embodied conversational agents. This framework is composed of a low-level communication platform, a set of communication API libraries, a high-level protocol as well as a reference starter toolkit for building ECAs. An XML based script language called GECA Scenario Markup Language (GSML) defining human-agent interactions and its execution component were developed to supplement GECA.

We showed GECA's usability by developing a variate of ECA systems. They include a multi-culture virtual tour guide agent and quiz agents. The first prototype of the quiz agent is actually deployed in public exhibitions for two years. It is then improved with two approaches to achieve participant attentiveness in a multi-participant configuration which is typical in public exhibitions. These two agents use video and audio information from the activity of the participants to determine the timings of their verbal and nonverbal actions respectively. These two quiz agents are then evaluated with questionnaires, a quantitative psychology method called GNAT and video analysis. The experiment results showed that by controlling the timings of actions can indeed improve the life-likeness of ECAs. Finally, a visual knowledge management system called Gallery is proposed for managing large-size collections of story-telling style content that can be presented by ECAs. In the evaluation experiments, it showed its effectiveness comparing to a well-known commercial image management application.

We would like to extend the framework to support the development of more sophisticated ECAs in the future, publish it when it is ready, and hope it can contribute to the research efforts in developing embodied conversational agents.

References

ACD Systems. (2004). *Acdsee photo manager.* WebSite. Available from http://www.acdsee.com/

Adobe Systems Inc. (2004). *Adobe photoshop album.* WebSite. Available from http://www.adobe.com/products/photoshopalbum/index.html

Alavi, M., & Leidner, D. E. (1999). Knowledge management systems: Issues, challenges, and benefits. *Communications of Association for Information Systems, 1*(7), 2–36.

A.L.I.C.E. AI Fnd. (2005). *Artificial Intelligence Markup Language (AIML).* Website. Available from http://www.alicebot.org (http://www.alicebot.org/TR/2005/WD-aiml/)

Anderson, J. (2000). *Learning and memory: an integrated approach* (2nd ed.). Wiley.

Andre, E., & Rist, T. (2001, March). Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems. *Knowledge-Based Systems, 14*(1-2), 3–13.

Apache Software Fnd. (2004). *Apache http server project.* Website. Available from http://httpd.apache.org/

Arafa, Y., & Mamdani, A. (2003). Scripting embodied agents behaviour with cml: character markup language. In *Proceedings of the 8th international conference on intelligent user interfaces* (pp. 313–316).

Ascension Tech. (2009). *MotionStar.* Website. Available from http://www.ascension-tech.com/realtime/MotionSTARWirelessLITE.php

Autodesk. (2009). *3ds Max.* Website. Available from http://usa.autodesk.com

Baylor, A. L., Rosenberg-Kima, R. B., & Plant, E. A. (2006). Interface agents as social

models: The impact of appearance on females' attitude toward engineering. In *Conference on human factors in computing systems (CHI'06).*

Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., et al. (2001). *Manifesto for agile software development.* Website. Available from `http://www.agilemanifesto.org/`

Becker, C., Kopp, S., & Wachsmuth, I. (2004). Simulating the emotion dynamics of a multimodal conversational agent. In *Proceedings on tutorial and research workshop on affective dialogue systems (ADS'04).*

Becker, C., Kopp, S., & Wachsmuth, I. (2007). Conversational informatics. In T. Nishida (Ed.), (pp. 49–67). John Wiley & Sons, Ltd.

Becker, C., Nakasone, A., Prendinger, H., Ishizuka, M., & Wachsmuth, I. (2005). Physiologically interactive gaming with the 3d agent max. In *International workshop on conversational informatics, in conj. with JSAI'05.*

Becker, C., Prendinger, H., Ishizuka, M., & Wachsmuth, I. (2005). Evaluating affective feedback of the 3d agent max in a competitive cards game. In *The first international conference on affective computing and intelligent interaction (ACII'05).*

Bederson, B. B. (2001). Photomesa: A zoomable image browser using quantum treemaps and bubblemaps. In *The proceedings of UIST'01* (pp. 71–80).

Bederson, B. B., Grosjean, J., & Meyer, J. (2004, August). Toolkit design for interactive structured graphics. *IEEE Transactions on Software Engineering, 30*(8), 535–546.

Bell, L., & Gustafson, J. (2003). Child and adult speaker adaptation during error resolution in a publicly available spoken dialogue system. In *Proceedings of eurospeech 2003* (pp. 613–616).

Boukricha, H., Becker, C., & Wachsmuth, I. (2007). Simulating empathy for the virtual human max. In *2nd international workshop on emotion and computing, in conj. with the german conference on artificial intelligence (KI'07),.*

Carolis, B. D., Rosis, F. de, Carofiglio, V., Pelachaud, C., & Poggi, I. (2001). Interactive information presentation by an embodied animated agent. In *International class workshop on information presentation and natural multi-modal dialogue.*

Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjlmsson, H., et al. (1999). Embodiment in conversational interfaces: Rea. In *The proceedings of*

CHI'99.

Cassell, J., Bickmore, T., Campbell, L., Vilhjalmsson, H., & Yan, H. (2000). Embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), (pp. 29–63). The MIT Press.

Cassell, J., Stocky, T., Bickmore, T., Gao, Y., Nakano, Y., Ryokai, K., et al. (2002). Mack: Media lab autonomous conversational kiosk. In *Proceedings of Imagina'02*.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. (Eds.). (2000). *Embodied conversational agents*. MIT Press.

Cerekovic, A., Huang, H.-H., Furukawa, T., Yamaoka, Y., Pandzic, I. S., Nishida, T., et al. (2008, August 4-29). Implementing a multiparty support in a tour guide system with an embodied conversational agent (ECA). In *The enterface2008 international workshop on multimodal interfaces.* Orsay, France.

Chang, C.-C., & Lin, C.-J. (2008). *LIBSVM – a library for support vector machines.* website. Available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

CMU. (2009). *Panda3d.* Website. Available from `http://www.panda3d.org/`

Combs, T. T. A., & Bederson, B. B. (1999). Does zooming improve image browsing. In *The proceedings of DL'99* (pp. pp130–137).

Costa, A., Pickering, M. J., & Sorace, A. (2008). Alignment in second language dialogue. *Language and Cognitive Processes*, *23*(4), 528–556.

Czerwinski, M., Dantzich, M., Robertson, G., Dziadosz, S., Tiernan, S., & Dantzich, M. van. (1999). The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3d. In *Proceedings of INTERACT'99* (pp. 163–170).

Egges, A., & Molet, T. (2004). Personalised real-time idle motion synthesis. In *Proceedings. 12th pacific conference on computer graphics and applications* (pp. 121–130).

Egges, A., & Visser, R. (2004). Example-based idle motions in a real-time application. In *Proceedings of captech workshop 2004.* Zermatt, Switzerland.

Eichner, T., Prendinger, H., Andre, E., & Ishizuka, M. (2007). Attentive presentation agent. In *Proceedings of the 7th international conference on intelligent virtual agents (IVA'07)* (pp. 283–295). Paris, France.

Ekman, P., Friesen, W. V., & Hager, J. C. (2002). *Facial action coding system (facs).*

Website. Available from http://www.face-and-emotion.com/dataface/facs/description.jsp

Gebhard, P., Schroder, M., Chaarafuelan, M., Endres, C., Kipp, M., Pammi, S., et al. (2008). Ideas4games: Building expressive virtual characters for computer games. In *Proceedings of the 8th international conference on intelligent virtual agents (IVA'08)* (pp. 426–440).

Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S., Morales, M., Werf, R. van der, et al. (2006). Virtual rapport. In *Proceedings of the 6th international conference on intelligent virtual agents (IVA'06)* (pp. 14–27). Marina del Rey, USA.

Gratch, J., Rickel, J., Andre, E., Cassell, J., Petajan, E., & Badler, N. (2002). Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, *17*(4), 54–63. Available from citeseer.ist.psu.edu/gratch02creating.html

Gustafson, J., & Bell, L. (2000). Speech technology on trial experiences from the august system. *Natural Language Engineering, Cambridge University Press*, *1*(1), 1–15.

Gustavsson, C., Beard, S., Strindlund, L., Huynh, Q., Wiknertz, E., Marriott, A., et al. (2001, October). Vhml (Working Draft v0.3 ed.) [Computer software manual]. Available from http://www.vhml.org/documents/VHML/2001/WD-VHML-20011021/

H-Anim WG. (2002). *Humanoid Animation (H-Anim)*. Website. Available from http://www.hanim.org

Hacker, B. A., Wankerl, T., Kiselev, A., Huang, H.-H., Merckel, L., Okada, S., et al. (2009, June). Incorporating intentional and emotional behaviors into a virtual human for better customer-engineer-interaction. In I. P. Zarko & B. Vrdoljak (Eds.), *10th international conference on telecommunications (ConTEL'09)* (pp. 163–170). Zagreb, Croatia: University of Zagreb.

Hall, E. T. (1992). *Beyond culture*. Peter Smith Publisher.

Hamiru.aqui. (2004). *70 japanese gestures - no language communication*. IBC Publishing.

Hori, K. (1994). A system for aiding creative concept formation. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(6), 882–894.

Hoya Corp. (2008). *Pentax VoiceText text-to-speech engine.* Available from http://voice.pentax.jp/

Iacobelli, F., & Cassell, J. (2007). Ethnic identity and engagement in embodied conversational agents. In *Proceedings of the 7th international conference on intelligent virtual agents (IVA'07)* (pp. 57–63). Paris, France: Springer.

IETF. (1992, March). *Rfc1305: Network time protocol (version 3) specification, implementation and analysis.* Website. Available from `http://tools.ietf.org/html/rfc1305`

IETF. (1996, October). *Rfc2030: Simple Network Time Protocol (SNTP) version 4 for IPv4, IPv6 and OSI.* Website. Available from `http://tools.ietf.org/html/rfc2030`

IETF. (2003, July). *Rfc3548: The base16, base32, and base64 data encodings.* Website. Available from `http://tools.ietf.org/html/rfc3548`

Intel Corp. (2006). *Open Computer Vision Library (OpenCV) 1.0.* Available from `http://sourceforge.net/projects/opencvlibrary/`

Ipšič, S., Žanert, J., & Ipšič, I. (2003, July). Speech recognition of croatian and slovenian weather forecast. In *Proceedings of 4th eurasip conference* (pp. 637–642).

Isbister, K. (2004). Agent culture: Human-agent interaction in a multicultural world. In S. Payr & R. Trappl (Eds.), (pp. 233–244). Lawrence Erlbaum Associates.

ISO/IEC JTC1. (1999). *Iso/iec jtc1/sc29/wg11, iso/iec 14496:1999, coding of audio, picture, multimedia and hypermedia information, n3056.*

ITU-T. (2002, May). *Network performance objectives for ip-based services* (Recommendation No. Y.1541).

Johnson, W. L., Vilhjalmsson, H., & Marsella, S. (2005). Serious games for language learning: How much game, how much ai? In *Proceedings of the 12th international conference on artificial intelligence in education.*

Kato, H. (n.d.). *Artoolkit.* http://www.hitl.washington.edu/artoolkit/. Available from `http://artoolkit.sourceforge.net/`

Kawakita, J. (1975). *The kj method. a scientific approach to problem solving.* Kawakita Research Institute.

Kendon, A. (2004). *Gesture: Visible actions as utterance.* Cambridge University Press.

Klesen, M., Kipp, M., Gebhard, P., & Rist, T. (2003, December). Staging exhibitions: methods and tools for modelling narrative structure to produce interactive performances with virtual actors. *Virtual Reality, 7*(1), 17–29.

Kopp, S., Allwood, J., Grammer, K., Ahlsen, E., & Stocksmeier, T. (2008). Modeling communication with robots and virtual humans. In I. Wachsmuth & G. Knoblich (Eds.), (pp. 18–37). Springer Berlin / Heidelberg.

Kopp, S., Bergmann, K., & Wachsmuch, I. (2008, March). Multimodal communication from multimodal thinking - towards an integrated model of speech and gesture production. *International Journal on Semantic Computing*, *2*(1).

Kopp, S., Gesellensetter, L., Kramer, N. C., & Wachsmuth, I. (2005). A conversational agent as museum guide - design and evaluation of a real-world application. In *Proceedings of the 5th international conference on intelligent virtual agents (IVA'05).* Kos, Greece.

Kopp, S., & Jung, B. (2000). An anthropomorphic assistant for virtual assembly: Max. In *Working notes workshop "communicative agents in intelligent environments", autonomous agents '00.*

Kopp, S., Jung, B., Lebmann, N., & Wachsmuth, I. (2003). Max - a multimodal assistant in virtual reality construction. *KI Künstliche Intelligenz*, *03*(4), 11–17.

Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., et al. (2006). Towards a common framework for multimodal generation: The behavior markup language. In *Proceedings of the 6th international conference on intelligent virtual agents (IVA'06)* (pp. 205–217). Marina del Rey, USA.

Kranstedt, A., Kopp, S., & Wachsmuth, I. (2002). *Murml: A multimodal utterance representation markup language for conversational agents* (Tech. Rep.). SFB 360 Situated Artifical Communicators, Universit 舩 Bielefeld.

Kshirsagar, S., Magnenat-Thalmann, N., Guye-Vuilleme, A., Thalmann, D., Kamyab, K., & Mamdani, E. (2002). Avatar markup language. In *Eigth eurographics workshop on virtual environments.*

Kubota, H., Kurohashi, S., & Nishida, T. (2004). Virtualized egos using knowledge cards. *Electronics and Communications in Japan*, *88*(1), 32–39.

Kubota, H., Nomura, S., Sumi, Y., & Nishida, T. (2007). Sustainable memory system using global and conical spaces. *Journal of Universal Computer Science*, *13*(2), 135–148.

Kuchinsky, A., Pering, C., Creech, M., Freeze, D., Serra, B., & Gwizdka, J. (1999). Fotofile: A consumer multimedia organization and retrieval system. In *The proceedings of CHI*

'99 (pp. 496–503).

Larsson, S., Berman, A., Gronqvist, L., & Kronlid, F. (n.d.). Trindikit 3.0 manual [Computer software manual]. (Trindi Deliverable D6.4)

Lieberman, H., Rosenzweig, E., & Singh, P. (2001). Aria: an agent for annotating and retrieving images. *IEEE Computer*, *34*(7), 57–61.

Loquendo Corp. (2008). *Loquendo ASR.* WebSite. Available from http://www.loquendo.com/en/

Mehrabian, A. (1996, December). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, *14*(4), 261–292.

Microsoft Corp. (2001). *Speech Application Programming Interface ver. 5.1.* Website. Available from http://www.microsoft.com/speech

mindmakers.org. (2005). *OpenAIR Protocol Specification 1.0.* Website. Available from http://www.mindmakers.org/openair/airPage.jsp (http://www.mindmakers.org/openair/airPage.jsp)

mindmakers.org. (2006). *SAIBA multimodal behavior generation framework.* Website. Available from http://wiki.mindmakers.org/projects:SAIBA:main

Mindmakers.org. (2008, August). *Behavior markup language (bml) version 1.0 (draft).* Website. Available from http://wiki.mindmakers.org/projects:bml:draft1.0

Morikawa, O., & Maesako, T. (1998). Hypermirror: Toward pleasant-to-use video mediated communication system. In *Proceedings of the 1998 acm conference on computer supported cooperative work (CSCW'98)* (pp. 149–158). New York, NY, USA: ACM Press.

MotionAnalysis Inc. (2009). *Mac 3d.* Website. Available from http://www.motionanalysis.com/html/animation/products.html

Nakano, Y., Okamoto, M., Kawahara, D., Li, Q., & Nishida, T. (2004). Converting text into agent animations: Assigning gestures to text. In *Proceedings of the human language technology conference (HLT-NAACL'04).*

Nakano, Y. I., Reinstein, G., Stocky, T., & Cassell, J. (2003). Towards a model of face-to-face grounding. In *Proceedings of the 41st annual meeting of the association for*

computational linguistics (ACL'03) (pp. 553–561).

Nakata, A., Kijima, H., & Sumi, Y. (2009). *iCorpusStudio.* Website. Available from http://www.ii.ist.i.kyoto-u.ac.jp/iCorpusStudio/index.html

Nass, C., Isbister, K., & Lee, E.-J. (2000). Embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), (pp. 374–402). The MIT Press.

NHK . (2009). *TV program Markup Language (TVML).* Website. Available from http://www.nhk.or.jp/strl/tvml/english/player2/index.html

Nishida, T. (Ed.). (2007). *Conversational informatics: An engineering approach.* John Wiley & Sons Inc.

Nosek, B. A., & Banaji, M. R. (2001). The go/no-go association task. *Social Cognition, 19*(6), 625–664.

Oka, K., & Sato, Y. (2005). Real-time modeling of a face deformation for 3d head pose estimation. In *Proc. ieee international workshop on analysis and modeling of faces and gestures (AMFG'05).*

Okazaki, N., Aya, S., Saeyor, S., & Ishizuka, M. (2002). A multimodal presentation markup language mpml-vr for a 3d virtual space. In *Workshop proc. (cd-rom) on virtual conversational characters: Applications, methods, and research challenges (in conjunction with HF'02 and OZCHI'02).* Melbourne, Australia.

Omron Corp. (2008). *OKAO Vision.* Website. Available from http://www.omron.com/r_d/coretech/vision/okao.html

Pandzic, I. S., & Forchheimer, R. (Eds.). (2002). *Mpeg-4 facial animation, the standard, implementation and applications.* John Wiley & Sons Inc.

Peic, R. (2003, July). A speech recognition algorithm based on the features of croatian language. In *Proceedings of the 4th eurasip conference* (pp. 613–618).

Pelachaud, C., Carofiglio, V., Rosis, F. de, & Poggi, I. (2002). Embodied contextual agent in information delivering application. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS'02)* (pp. 758–765).

PhaseSpace Inc. (2009). *PhaseSpace motion capture.* Website. Available from http://www.phasespace.com/

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue.

Behavioral and brain sciences, 27, 169–226.

Pickering, M. J., & Garrod, S. (2006). Alignment as the basis for successful communication. *Research on Language and Computation, 4*, 203–228.

Platt, J., Czerwinski, M., & Field, B. (2002). *Phototoc: Automatic clustering for browsing personal photographs* (Tech. Rep. No. MSR-TR-2002-17). Microsoft Research.

Pokahr, A., Braubach, L., & Lamersdorf, W. (2005, September). Jadex: A bdi reasoning engine. In R. Bordini, M. Dastani, J. Dix, & A. E. F. Seghrouchni (Eds.), *Multi-agent programming* (pp. 149–174). Springer Science+Business Media Inc.

Prendinger, H., Descamps, S., & Ishizuka, M. (2002). Scripting affective communication with life-like characters in web-based interaction systems. *Applied Artificial Intelligence, 16*(7–8), 519–553.

Prendinger, H., & Ishizuka, M. (Eds.). (2004). *Life-like characters - tools, affective functions, and applications*. Springer.

Rehm, M. (2008, May). "she is just stupid"-analyzing user-agent interactions in emotional game situations. *Interacting with Computers, 20*(3), 311–325.

Rehm, M., & Andre, E. (2005, September). Where do they look? gaze behaviors of multiple users interacting with an embodied conversational agent. In *Proceedings of the 5th international conference on intelligent virtual agents (IVA'05)*. Kos, Greece.

Rehm, M., Andre, E., Bee, N., Endrass, B., Wissner, M., Nakano, Y., et al. (2007, July). The cube-g approach - coaching culture-specific nonverbal behavior by virtual agents. In *the 38th conference of the international simulation and gaming association (ISAGA'07)*. Nijmegen, New Zealand.

Rehm, M., Andre, E., & Wissner, M. (2005). Gamble v2.0 - social interactions with multiple users. In *The 4th international joint conference on autonomous agents and multiagent systems (AAMAS'05)* (pp. 145–146).

Rehm, M., Bee, N., Endrass, B., Wissner, M., & Andre, E. (2007). Too close for comfort? In *Proceedings of the international workshop on human-centered multimedia, acm multimedia (2007)*.

Rehm, M., Gruneberg, F., Nakano, Y., Lipi, A. A., Yamaoka, Y., & Huang, H.-H. (2008, January). Creating a standardized corpus of multimodal interactions for enculturating conversational interfaces. In *Workshop on enculturating conversational interfaces by*

socio-cultural aspects of communication, 2008 international conference on intelligent user interfaces (IUI'08). Canary Islands, Spain.

Rehm, M., Nakano, Y., Andre, E., & Nishida, T. (2008, September). Culture-specific first meeting encounters between virtual agents. In H. Prendinger, J. Lester, & M. Ishizuka (Eds.), *Proceedings of the 8th international conference on intelligent virtual agents (IVA'08)* (pp. 223–236). Tokyo, Japan.

Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D., & Dantzich, M. van. (1998). Data mountain: Using spatial memory for document management. In *The proceedings of UIST '98* (pp. 153–162).

Robinson, S., Traum, D., Ittycheriah, M., & Henderer, J. (2008). What would you ask a conversational agent? observations of human-agent dialogues in a museum setting. In *Language resources and evaluation conference (lrec).*

Rodden, K. (1999). How do people organize their photographs? In *The proceedings of the bcs irsg colloquium, eletronic workshops in computing.*

Rodden, K., & Wood, K. (2003). How do people manage their digital photographs? In *The proceedings of CHI'03* (pp. pp409–416).

Rosis, F. de, Pelachaud, C., & Poggi, I. (2004). Agent culture: Human-agent interaction in a multicultural world. In S. Payr & R. Trappl (Eds.), (pp. 75–105). Lawrence Erlbaum Associates.

Shibata, H., & Hori, K. (2002). A system to support long-term creative thinking in daily life and its evaluation. In *Proceedings of creativity and cognition* (pp. 142–149).

Solomon, S., Lent, M. van, Core, M., Carpenter, P., & Rosenberg, M. (2008). A language for modeling cultural norms, biases and stereotypes for human behavior models. In *Proceedings of the 17th conference on behavior representation in modeling and simulation (BRIMS'08).*

Sumi, Y., Hori, K., & Ohsuga, S. (1997). Computer-aided thinking by mapping text-objects into metric spaces. *Artificial Intelligence*, *91*(1), 71–84.

Sun Microsystems. (2004a). *Java platform enterprise edition.* Website. Available from `http://java.sun.com/javaee/reference/`

Sun Microsystems. (2004b). *MySQL.* Website. Available from `http://www.mysql.com/`

Thiebaux, M., Marshall, A. N., Marsella, S., & Kallmann, M. (2008). Smartbody: Behavior

realization for embodied conversational agents. In *The 7th international conference of autonomous agents and multiagent systems (AAMAS'08).* Estorial, Portugal.

Traum, D. (2003). Issues in multiparty dialogues. In *Advances in agent communication, international workshop on agent communication languages (ACL'03)* (pp. 201–211).

Traum, D., Bos, J., Cooper, R., Larsson, S., Lewin, I., Matheson, C., et al. (1999). *A model of dialogue moves and information state revision.* Available from http://www.ling.gu.se/projekt/trindi/private/deliverables/D2.1/D2.1.pdf

Traum, D., Rickel, J., Gratch, J., & Marsella, S. (2003). Negotiation over tasks in hybrid human-agent teams for simulation-based training. In *The 2nd international conference of autonomous agents and multiagent systems (AAMAS'03).* Melbourne, Australia.

Vertegaal, R., Slagter, R., Veer, G. van der, & Nijholt, A. (2001). 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* (pp. 301–308).

Vilhjálmsson, H., Cantelmo, N., Cassell, J., Chafai, N. E., Kipp, M., Kopp, S., et al. (2007). The behavior markup language: Recent developments and challenges. In *Proceedings of the 7th international conference on intelligent virtual agents (IVA'07)* (pp. 99–111). Paris, France: Springer.

Visage Technologies AB. (2008). *Visage|SDK.* Website. Available from http://www.visagetechnologies.com (http://www.visagetechnologies.com)

W3C. (2004). *Emma: Extensible multimodal annotation markup language.* http://www.w3.org/TR/emma/. Available from http://www.w3.org/TR/emma/

Web3D Consortium. (1997). *The virtual reality modeling language (VRML).* Website. Available from http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/

Weizenbaum, J. (1966, January). Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Young, P. A. (2008). Integrating culture in the design of icts. *British Journal of Educational Technology*, *39*(1), 6–17.

Zoric, G., & Pandzic, I. S. (2005, July 6-8). A real-time language independent lip synchronization method using a genetic algorithm. In *the proceedings of ICME'05*. Amsterdam, The Netherland.

Publications

I. First Author Papers

A. International Journal Papers

1. H.-H. Huang, Y. Sumi, and T. Nishida. Personal image repositories as externalized memory spaces. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 10(2):169–180, April 2006. IOS Press.

2. H.-H. Huang, A. Cerekovic, K. Tarasenko, V. Levacic, G. Zoric, I. S. Pandzic, Y. Nakano, and T. Nishida. An agent based multicultural tour guide system with nonverbal user interface. *International Journal on Multimodal User Interfaces*, 1(1):41–48, April 2008. Springer Berlin.

3. H.-H. Huang, A. Cerekovic, K. Tarasenko, V. Levacic, G. Zoric, I. S. Pandzic, Y. Nakano, and T. Nishida. Integrating embodied conversational agent components with a generic framework. *Multiagent and Grid Systems*, 4(4):371–386, January 2009. IOS Press.

4. H.-H. Huang, A. Cerekovic, I. S. Pandzic, Y. Nakano, and T. Nishida. Toward a multi-culture adaptive virtual tour guide agent with a modular approach. *AI & Society Journal of Knowledge, Culture and Communication*, 24(3):225–235, October 2009. Springer London.

B. International Conference Papers

1. H.-H. Huang, Y. Sumi, and T. Nishida. Galley: in support of human memory. In *Proceedings of 8th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'04), Lecture Notes in Computer Science*, volume 3213/2004, pages 357–363, Wellington, New Zealand, September 2004. Springer Berlin.

2. H.-H. Huang, A. Cerekovic, K. Tarasenko, V. Levacic, G. Zoric, M. Treumuth, I. S. Pandzic, Y. Nakano, and T. Nishida. An agent based multicultural user interface in a customer service application. In *The eNTERFACE'06 International Workshop on Multimodal Interfaces*, Dubrovnik, Croatia, August 2006.

3. H.-H. Huang, T. Masuda, A. Cerekovic, K. Tarasenko, I. S. Pandzic, Y. Nakano, and T. Nishida. Toward a universal platform for integrating embodied conversational agent components. In *Proceedings of 10th International Conference on Knowledge- Based and Intelligent Information & Engineering Systems (KES'06), Lecture Notes in Computer Science*, volume 4252/2006, pages 220–226, Bournemouth, UK, October 2006. Springer Berlin.

4. H.-H. Huang, A. Cerekovic, I. S. Pandzic, Y. Nakano, and T. Nishida. A script driven multimodal embodied conversational agent based on a generic framework. In *C. Pelachaud, J.-C. Martin, E. Andre, G. Chollet, K. Karpouzis, and D. Pele, editors, 7th International Conference on Intelligent Virtual Agents (IVA'07)*, volume 4722 of *Lecture Notes in Artificial Intelligence*, pages 381–382, Paris, France, September 2007. Springer Berlin.

5. H.-H. Huang, T. Inoue, A. Cerekovic, I. S. Pandzic, Y. Nakano, and T. Nishida. A quiz game console based on a generic embodied conversational agent framework. In C. Pelachaud, J.-C. Martin, E. Andre, G. Chollet, K. Karpouzis, and D. Pele, editors, *7th International Conference on Intelligent Virtual Agents (IVA'07)*, volume 4722 of *Lecture Notes in Artificial Intelligence*, pages 383–384, Paris, France, September 2007. Springer Berlin.

6. H.-H. Huang, A. Cerekovic, I. S. Pandzic, Y. Nakano, and T. Nishida. Scripting human-agent interactions in a generic ECA framework. In R. Ellis, T. Allen, and M. Petridis, editors, *Applications and Innovations in Intelligent Systems XV, Proceedings of AI'07, the 27th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 103–115, Cambridge, UK, December 2007. Springer London.

7. H.-H. Huang, A. Cerekovic, I. Pandzic, Y. Nakano, and T. Nishida. Toward a culture adaptive conversational agent with a modularized approach. In *Workshop on Enculturating Conversational Interfaces by Socio-cultural Aspects of Communication, 2008 International Conference on Intelligent User Interfaces (IUI'08)*, January 2008.

8. H.-H. Huang, A. Cerekovic, Y. Nakano, I. S. Pandzic, and T. Nishida. The design of a generic framework for integrating ECA components. In L. Padgham, D. Parkes, and J. P. Muller, editors, *The 7th International Conference of Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 128–135, Estorial, Portugal, May 2008. Inesc-Id.

9. H.-H. Huang, T. Furukawa, H. Ohashi, Y. Ohmoto, and T. Nishida. Toward a virtual quiz agent who interacts with user groups. In *the 7th International Workshop on Social Intelligence Design (SID'08)*, Puerto Rico, December 2008.

10. H.-H. Huang, T. Furukawa, H. Ohashi, A. Cerekovic, Y. Yamaoka, I. S. Pandzic, Y. Nakano, and T. Nishida. Communicating with multiple users for embodied conversational agents. In I. P. Zarko and B. Vrdoljak, editors, *10th International Conference on Telecommunications (ConTEL'09)*, pages 155–162, Zagreb, Croatia, June 2009. IEEE, University of Zagreb.

11. H.-H. Huang, H. Kubota, and T. Nishida. Toward the knowledge circulation between institutions and public audiences with virtual agents. In B.-C. Chien and T.-P. Hong, editors, *Opportunities and Challenges for Next-Generation Applied Intelligence, the 22nd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE'09)*, volume 214/2009 of Studies in Computational Intelligence, pages 219–224, Tainan, Taiwan, June 2009. Springer Berlin.

12. H.-H. Huang, T. Furukawa, H. Ohashi, A. Cerekovic, Y. Yamaoka, I. S. Pandzic, Y. Nakano, and T. Nishida. The lessons learned in developing multi-user attentive quiz agents. In *9th International Conference on Intelligent Virtual Agents (IVA'09)*, volume 5571 of *Lecture Notes in Artificial Intelligence*, pages 166–173, Amsterdam, Netherlands, September 2009. Springer Berlin.

C. Domestic Conference Papers

1. 黄 宏軒, 角 康之, 西田 豊明. Galley: 人間記憶支援システム. 日本人工知能学会第 18 回全国大会, 石川県金沢市, 2004 年 6 月.

2. 黄 宏軒, 野村 聡史, 角 康之, 西田 豊明. 身体感覚によるコンテンツ整理を目指して. 日本人工知能学会第 19 回全国大会, 福岡県北九州市, 2005 年 6 月.

3. 黄 宏軒, A. Cerekovic, I. Pandzic, 中野 有紀子, 西田 豊明. Generic ECA フレームワークにおけるヒューマンエージェントインタラクションの記述. 合同エージェントワークショップ＆シンポジウム 2007(JAWS2007), 沖縄県宜野湾市, 2007 年 10 月.

II. Co-authored Papers

A. Domestic Journal Papers

1. 曲山 幸生, 久保田 秀和, 黄. 宏軒, 金井 二三子, 西田 豊明. 食総研における新しい研究成果発信方法の活用ー消費者を重視したコミュニケーションを目指してー. 情報管理, 51(2):116–128, 2008 年 5 月.

B. International Conference Papers

1. K. Okamoto, N. Y., M. Okamoto, H.-H Huang, and T. Nishida. Producing camera-work in CG movies based on a cognitive shot transition model. In *Proceedings of 9th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'05)*, Lecture Notes in Computer Science, volume 3683/2005,

pages 848–854, Melbourne, Australia, September 2005. Springer Berlin.

2. A. Cerekovic, H.-H. Huang, G. Zoric, K. Tarasenko, V. Levacic, I. S. Pandzic, Y. Nakano, and T. Nishida. Towards an embodied conversational agent talking in croatian. In Z. Car and M. Kusek, editors, *Proceedings of the 9th International Conference on Telecommunications (ConTEL'07)*, pages 41–48, Zagreb, Croatia, June 2007. IEEE, University of Zagreb.

3. M. Rehm, E. Andre, N. Bee, B. Endrass, M. Wissner, Y. Nakano, T. Nishida, and H.-H. Huang. The CUBE-G approach - coaching culture-specific nonverbal behavior by virtual agents. In *the 38th Conference of the International Simulation and Gaming Association (ISAGA'07)*, Nijmegen, New Zealand, July 2007.

4. A. Cerekovic, H.-H. Huang, I. S. Pandzic, Y. Nakano, and T. Nishida. Toward a multicultural eca tour guide system. In C. Pelachaud, J.-C. Martin, E. Andre, G. Chollet, K. Karpouzis, and D. Pele, editors, *7th International Conference on Intelligent Virtual Agents (IVA'07)*, volume 4722 of *Lecture Notes in Artificial Intelligence*, pages 364–365, Paris, France, September 2007. Springer Berlin.

5. M. Rehm, F. Gruneberg, Y. Nakano, A. A. Lipi, Y. Yamaoka, and H.-H. Huang. Creating a standardized corpus of multimodal interactions for enculturating conversational interfaces. In *Workshop on Enculturating Conversational Interfaces by Socio-cultural Aspects of Communication, 2008 International Conference on Intelligent User Interfaces (IUI'08)*, Canary Islands, Spain, January 2008.

6. A. Cerekovic, H.-H. Huang, T. Furukawa, Y. Yamaoka, I. S. Pandzic, T. Nishida, and Y. Nakano. Implementing a multiparty support in a tour guide system with an embodied conversational agent (ECA). In *The eNTERFACE'08 International Workshop on Multimodal Interfaces*, Orsay, France, August 2008.

7. B. A. Hacker, T.Wankerl, A. Kiselev, H.-H. Huang, L. Merckel, S. Okada, J. Schlichter, N. Abdikkev, and T. Nishida. Incorporating intentional and emotional behaviors into a virtual human for better customer-engineer-interaction. In I. P. Zarko and B. Vrdoljak, editors, *10th International Conference on Telecommunications (ConTEL'09)*, pages 163–170, Zagreb, Croatia, June 2009. IEEE, University of Zagreb.

8. A. Cerekovic, H.-H. Huang, T. Furukawa, Y. Yamaoka, I. S. Pandzic, T. Nishida, and Y. Nakano. Implementing a multi-user tour guide system with an embodied conversational agent. In *the 5th International Conference on Active Media Technology (AMT'09)*, Beijing, China, October 2009.

Appendix A

GECA Scenario Markup Language (GSML) Reference

The GECA scenario markup language shares the same basic idea of AIML. To write an AIML script, the agent creator defines a set of *pattern-template* pairs to describe the possible interactions between an agent and its human user. The agent says the utterance described in a template in responding to a user's utterance described in a pattern. `Category` is a container of exactly one pattern-template pair. An AIML script is a list of categories that compose the agent's knowledge.

GECA Scenario Markup Language further extends the basic idea of AIML to fit the needs to build an ECA which can interact its user in verbal and non-verbal modalities. One scenario script defines an interaction scenario between the agent and the human user. A scenario can contain multiple scenes while each scene presents a location in the virtual world and is decorated by a background image. In an individual scene, the conversation between the agent and the user is modeled by one or more conversational states. For example, consider a guide kiosk application of a museum; the guide agent stands in front of the entrance of the museum where it can guide the human user. Then a picture of the entrance of the museum can be the initial scene and "greeting," "ask the user which floor to go" conversational states can be used in this scene. The Scene-State-Category hierarchy limits the range of available responds into a conversational state and prevents the problem that an unexpected template may be triggered in AIML agent which practically has only one conversational

state. Further, in GECA Scenario ML, templates can be set up to be triggered right away when the conversation transfer into a new state without a user utterance.

In GECA Scenario ML, patterns and templates are extended to be able to describe non-verbal behaviors of agent and human user in addition to speech. `Action` tags that specify face or body animations (e.g. lip animation or non-verbal behaviors) can be inserted into the utterances of the agent, the timing information is specified by the position of the `Action` tags in the utterance texts. The `Perception` tags can be inserted inside the `Pattern` tags then the corresponding template will be triggered if the user does that non-verbal behavior. However, the order and combination of multiple perceptions and their relationship with a recognized speech is an issue that has to be solved in the future. Further, areas of the background image can be defined by `Object` elements and can be referenced (e.g. pointed at or gazed at) by the user during the conversation.

A.1 Complete GSML Document Type Definition (DTD)

```
<!ELEMENT Scenario (Scene+)>
<!ATTLIST Scenario Version       NUMBER   #REQUIRED
                   IitialScene   NAME     #REQUIRED>

<!ELEMENT Scene (State+ & Objects?)>
<!ATTLIST Scene ID             NAME     #REQUIRED
                InitialState   NAME     #IMPLIED
                X              NUMBER   #IMPLIED
                Y              NUMBER   #IMPLIED>

<!ELEMENT Objects (Object+)>

<!ELEMENT Object>
<!ATTLIST Object ID      NAME     #REQUIRED
                 X       NUMBER   #REQUIRED
                 Y       NUMBER   #REQUIRED
```

```
                Width   NUMBER   #REQUIRED
                Height  NUMBER   #REQUIRED>

<!ELEMENT State (Category* | InitialCategory?)>
<!ATTLIST State ID         Name    #REQUIRED
                Language   CDATA   "English">

<!ELEMENT Category (Pattern, Template)>

<!ELEMENT InitialCategory (Template)>

<!ELEMENT Pattern (CDATA | Perception*)>

<!ELEMENT Template (CDATA | Action* | Transition*)>

<!ELEMENT Perception (EMPTY)>
<!ATTLIST Perception Type     CDATA   #REQUIRED
                     Target   NAME    #IMPLIED>

<!ELEMENT Action (EMPTY)>
<!ATTLIST Action Type         NAME      #REQUIRED
                 SubType      NAME      #IMPLIED
                 Delay        NUMBER    "0"
                 Duration     NUMBER    "0"
                 Intensity    NUMBER    "0"
                 X            NUMBER    #IMPLIED
                 Y            NUMBER    #IMPLIED
                 Z            NUMBER    #IMPLIED
                 Direction    #PCDATA   #IMPLIED
                 Trajectory   (Linear | Sinusoidal | Oscillation)
                              "Sinusoidal"
```

```
                    Sync          (WithNext | BeforeNext |
                                  PauseSpeaking)   "WithNext">

<!ELEMENT Transition (EMPTY)>
<!ATTLIST Transition State   NAME   #REQUIRED
                     Scene   NAME   #IMPLIED>
```

A.2 Extended GSML DTD

* Only the differences from GSML are listed

```
<!ELEMENT Scenario (Scene+ & Information & GlobalState?)>
<!ATTLIST Scenario Version       NUMBER   #REQUIRED
                   IitialScene   NAME     #REQUIRED>

<!ELEMENT GlobalState (Category+)>

<!ELEMENT Inormation (Variable+)>

<!ELEMENT Variable (EMPTY)>
<!ATTLIST Variable Name      NAME                 #REQUIRED
                   Type      (Integer | String)   "String"
                   Default   #PCDATA              #REQUIRED>

<!ELEMENT Pattern (CDATA | Predicate*)>

<!ELEMENT Predicate (Argument*)>
<!ATTLIST Predicate Function   NAME   #REQUIRED>

<!ELEMENT Argument (EMPTY)>
<!ATTLIST Argument Value   CDATA   #REQUIRED>
```

```
<!ELEMENT Template (CDATA | Action* | Transition* | Effect*)>

<!ELEMENT Effect Function   NAME   #REQUIRED>
```

A.3 GSML Element Reference

Scenario Element

The root element of a GECA scenario script. There is exactly one `Scenario` element in one script. *Containing element:* at least one `Scene` element

Attribute	Description	Type	Number	Default
Version	The version of GECA Scenario ML. The scenario executing component reads the value of this attribute to determine how to execute this script.	Numeric	1	n/a
InitialScene	The ID of the initial scene.	Text	1	n/a

Scene Element

This element describes a specific scene that distinguishes to the other scenes by a background image. *Containing Element:* at least one `State` element

Attribute	Description	Type	Number	Default
ID	The ID of this scene. Note: scene IDs must be unique and are related to background image settings	Text	1	n/a
InitialState	The ID of the initial state of this scene. Note: the initial scene of the whole scenario must have an `InitialState` while the other scenes do not need to have one	Text	0 or 1	n/a
X	The width of the background image in pixels	Numeric	0 or 1	n/a
Y	The height of the background image in pixels	Numeric	0 or 1	n/a

Objects Element

This element is a container element of `Object` elements. *Containing Element:* at least one `Object` element

Attribute	Description	Type	Number	Default
n/a	n/a	n/a	n/a	n/a

Object Element

`Object` element defines a 2D area in the background image to present an object that can be referenced by the human user. *Containing Element:* none

Attribute	Description	Type	Number	Default
ID	The ID of this object Note: object IDs must be unique in the scope of a scene	Text	1	n/a
X	The X coordinate of the origin of the area presenting this object	Numeric	1	n/a
Y	The Y coordinate of the origin of the area presenting this object	Numeric	1	n/a
Width	The width of the area presenting this object	Numeric	1	n/a
Height	The height of the area presenting this object	Numeric	1	n/a

State Element

This element describes a state of human-agent conversation within one scene. *Containing Element:* at least one `Category` or `InitialCategory` element, at most one `InitialCategory` element

Attribute	Description	Type	Number	Default
ID	The ID of this conversational state. Note: state IDs must be unique in the scope of a scene	Text	1	n/a
Language	The language that is used inside this state, e.g. Japanese, English, Croatian, etc.	Text	0 or 1	English

Category Element

This element represents one conversation between the human user and the agent. *Containing Element:* exactly one `Pattern` element and exactly one `Template` element

Attribute	Description	Type	Number	Default
n/a	n/a	n/a	n/a	n/a

InitialCategory Element

This element represents the same meaning as normal `Category` elements except its `Template` will be initiated directly when the conversation between the human user and the agent get into the state it belongs to. Note: there is no `Pattern` element inside an `InitialCategory`. *Containing Element:* exactly one `Template` element

Attribute		Description	Type	Number	Default
n/a	n/a		n/a	n/a	n/a

Pattern Element

This element describes a pattern which will be used by the interpreter to match the verbal and nonverbal inputs from the human user. *Containing element:* none, but it contains a text string that is an utterance spoken by the human user as well as a description of the user's non-verbal behaviors described by a `Perception` element.

Attribute		Description	Type	Number	Default
n/a	n/a		n/a	n/a	n/a

Perception Element

`Perception` element presents a nonverbal behavior performed by the human user. Note: Current implementation of the scenario component responds to one and just one `Perception` element in each `Pattern` element. The order and combination of multiple perceptions and their relationship with a recognized speech is an issue that has to be solved. *Containing element:* at least one `Scene` element

Attribute	Description	Type	Number	Default
Type	The type of the non-verbal behavior performed by the user. Currently the available types include: "pointing"	Enum.	1	n/a
Target	This attribute specifies a target of the user's behavior if it is available. In the case of type, "pointing", it means the object ID which is pointed at by the user.	Text	0 or 1	n/a

Template Element

The element describes the behaviors that will be done by the agent to response to a pattern. *Containing element:* the agent's utterance that is in response to a patter as a text string as well as zero or more non-verbal behaviors that are described by `Action` elements. Besides, zero or one `Transition` element can be contained inside a `Template` element.

Attribute	Description	Type	Number	Default
n/a	n/a	n/a	n/a	n/a

Action Element

This element describes a non-verbal behavior of the agent. *Containing element:* none

Attribute	Description	Type	Number	Default
Type	The name of this action	Enum.	1	NULL
SubType	A supplement of the `Type` attribute	Text	0 or 1	NULL
Delay	The delay before actually playing this action when the player meets it. Represented in integer value and the unit is millisecond	Numeric	0 or 1	0
Duration	The duration for the agent to perform this action. Represented in integer value and the unit is millisecond	Numeric	1	0
Intensity	The intensity of this action. The valid values are in integer and the meaning of the values dependent on the Type attribute	Numeric	0 or 1	0
X	The X coordinate	Numeric	0 or 1	n/a
Y	The Y coordinate	Numeric	0 or 1	n/a
Z	The Z coordinate	Numeric	0 or 1	n/a
Direction	The direction of this action. The meaning and thus the possible and valid values depend on the `Type` attribute	Numeric	0 or 1	n/a
Trajectory	This attribute stands for the dynamics of this action. The possible values are: “Linear”, “Sinusoidal”, and “Oscillation”	Enum.	0 or 1	Sinusoidal
Sync	This attribute specifies the temporal relationship between the actions in an utterance. There are three possible values: “WithNext,” “BeforeNext,” and “PauseSpeaking.” Stands for do not wait for this action, wait for this action to end, and pause TTS while executing this action, respectively	Enum.	0 or 1	WithNext

Transition Element

This element represents a state transition. *Containing element:* none

Attribute	Description	Type	Number	Default
State	The target state	Text	1	n/a
Scene	The target scene. If this attribute is absent, it means that the target state is in the same scene as the source state	Text	0 or 1	n/a

A.3.1 Available Routine-generated Actions

We are not going to specify all of the actions that should be able to be performed by a GECA agent and leave the action set as application dependent. The following table lists the agent actions that have been implemented in the visage player. Note that the contrast gestures planned in the eNTERFACE project are not implemented yet. The actions with the "*" mark have their unique meanings in all GECA compatible animation players and should not be overrode.

Type	SubType	Direction	Intensity	Description
pointing		left, leftUp, right, rightUp, rightForward, leftFoward, backH, backE		The agent points in the directions that semantically correspond to the values defined for this attribute. In future, we plan to use the coordinates on the screen to which the agent should point. Thus using direction instead of the coordinates is a temporarily solution. "backE" and "backH" values represent variations of gestures with the elbow bent
banzai				The Japanese banzai gesture to show happiness

Type	SubType	Direction	Intensity	Description
bow			1-3	1 corresponds to a shallow bow, using only head; 2-is a deeper bow, very frequently used by Japanese people in a daily conversations, 3-corresponds to a very polite bow, showing a high respect to the listener
invite	Croatian, Japanese			The “invite” action of the “Croatian” subtype is waving upwards and then backwards with the left hand, a somewhat informal emblem gesture meaning inviting. The action of the subtype “Japanese” has not been implemented yet
handsCrossed				This is an emblem Japanese gesture, meaning that something is not allowed. The hands are crossed in front of the lower part of the chest
nodding				The action meaning both in Croatian and Japanese agreement, consent
shakeHead				The action meaning both in Croatian and Japanese negation or disapproval

Type	SubType	Direction	Intensity	Description
extend				This action means right arm extended with the palm open and oriented upwards. The meaning in the Japanese culture is “wait please.” At this moment, “extend” means extending the right arm. In the future we might need extending the left arm as well. Thus, the subtype attribute might be introduced with the “left/right” as possible values
wave				This action means oscillating right hand waving. Used in combination with the “extend” action as part of the Japanese gesture meaning “No. This is not true.” At this moment, “wave” means waving with the right hand. In the future we might need waving with the left hand as well. Thus the subtype attribute might be introduced with the “left/right” as possible values
expression	smile		1-2	Make the character to perform “smile” expression
	sad		1-2	Make the character to perform “sad” expression
	angry		1-2	Make the character to perform “angry” expression
	fearful			Make the character to perform “fearful” expression

Type	SubType	Direction	Intensity	Description
	surprised			Make the character to perform "surprised" expression
walking				Make the agent walk to the destination specified by X, Y, Z attributes
beat	a-e			Waving spontaneous gestures with either one or both arms, used by the CAST engine
contrast	a-c			Waving spontaneous gestures with either one or both arms, used by the BEAT engine
warning				An emblem gesture meaning danger: the elbow is bent and the hand is raised. In future, the finger feature needs to be implemented, i.e. the pointing finger only pointing upwards
playTrack	A track specifying string			This action type indicates that the animation player to play an animation track which is identified by the string in the attribute "SubType." The meaning of the value of SubType is animation player dependent. It can be a file name or an identifier to invoke an animation programmed in the player
turning				turn the agent 's whole body to face the direction (X, Z)
turnHead		left		turn the agent's head to left direction

Type	SubType	Direction	Intensity	Description
		right		turn the agent's head to right direction
		forward		turn the agent's head to the front
		leftFoward		
		rightForward		
position		forward, backward, left, right		Make the character to stand precisely at (X, Y, Z) and face to the direction specified by direction parameter
aruku				Make the agent to walk to (X, Y, Z)
point				Make the agent to point at (X, Y, Z) with her finger