Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Humanized User Interfaces

Doctoral Thesis

Ladislav Kunc

Prague, 09 / 2013

Ph.D. programme: Electrical Engineering and Information Technology
Branch of study: Information Science and Computer Engineering

Supervisor: Pavel Slavík

Acknowledgements

Firstly, I would like to thank to my supervisor *Pavel Slavík*. He have been always available for discussions and helped to move this work forward.

I owe many thanks to *Jan Kleindienst*, the head of Prague R&D Lab at IBM Czech Republic. Honza, it is a great pleasure to work with you and the whole research team. You came with an ECA idea and have encouraged and supported me to keep working on it.

I would also like to thank to my collaborators *Jan Macek* and *Jan Cuřín*. They helped me to get the ECA environment to some Netcarity EU project applications. This allowed usability testing the ECA with real users.

There are many collegues from faculty or from IBM Prague R&D Lab that were very helpful and discussed thesis problems with me, especially *Martin Labský*, *Tomáš Macek*, *Zdeněk Míkovec*.

Thanks goes also to faculty students and other people who participated in various usability tests or used the ECA platform as application developers.

Finally, my biggest thanks belongs to my family that always supported me during my studies and thesis writing.

This research has been partially supported by IBM Czech Republic.

This research has been partially supported by the European Commision under IST FP6 integrated project Netcarity, contract number IST-2006-045508.

This research has been partially supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics).

Contents

Humanized User Interfaces

Ladislav Kunc
kuncladi@fel.cvut.cz
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, Czech Republic

Abstract

Embodied Conversational Agent (ECA) is the user interface metaphor that allows to naturally communicate information during human-computer interaction. ECA represents a rich channel for conveying both verbal and non-verbal messages. The interaction logic and the communication including voice, gesture, emotion, text, video, etc. are driven by dialog and conversational application. In this work we present three main contribution areas. First, we introduce complex ECA platform that combines the presentation and interaction layer of an application. We address a problem of combining "classical" UI with ECA to create a rich multimodal output. Ontologies are used to enrich conversational interaction logic of the system and the user. Further, we analyze possibilities of dialog interaction strategies improvement in the area of human-computer turn-taking. New visual turn-yielding cues for ECA are implemented and compared. The mechanism of turn-yielding uses capabilities of ECA to convey proper signals in the right places of conversation. Specifically, we explore two hypotheses. In the first one, we try to find out whether using more turn-yielding cues before a dialog transition increases the probability of distinguishing the turn-yielding. The second hypothesis leads to verification whether there is a difference between strength of visual and vocal turn-yielding cues. The last contribution area consists of multiple ECA platform experiments with human participants. In the case of McGurk ECA test, the experimental procedure is extended to address the issue of human participants with corrected vision. The benefits of our solutions are shown on interactive talking head applications, especially in an interactive game called Billionaire.

Keywords

Human Computer Interaction, Multimodal systems, Dialog, Ontology, Embodied Conversational Agents, Turn-taking

The work presented in this thesis have been formed in years 2006 – 2013. It shows fruitful cooperation between two research institutes, Department of Computer Graphics and Interaction at Czech Technical University and Prague R&D Lab at IBM Czech Republic. I have been privileged to be part of both outstanding research groups. Rich discussions with my colleagues helped to develop ideas presented in this work.

1 Introduction

Our everyday human lives are tightly connected to computers. This connection is even larger due to the existence of various embedded appliances as mobile phones, PDAs, tablets or public kiosk computers. Therefore the human-computer interaction field becomes more important as these new appliances penetrate more and more people's lives. Tasks that computers solve are relatively broad and some are complex ones. User interfaces (UI) impose big demands on user cognitive abilities. Some UIs are designed for people who use them as a secondary task in a specific context (like in-car UIs – the primary task is driving) [160].

Humans use spoken language as a main communication mean. When participants of a conversation see each other they add various gestures to the messages, e.g. eye movements, facial expressions, body postures. Humanizing machine user interfaces means to add some form of an agent that will express human behavior. This thesis explores Embodied Conversational Agents (ECA) user interface metaphor. Applications can convey rich verbal and non-verbal messages to a user through ECAs. This work describes building of a complex ECA platform for human-like interaction and investigates various visual and nonverbal gestures, especially turn-taking. An integral part of this thesis are evaluations done with users which helped to shape the whole platform from the point of view of application developers and ECA interface users.

To make UIs more usable, designers should know at least three things of UI design [165]:

- *The user* who interacts with the system.
- *The system* that is being designed, functions of system and its use.
- *The interaction* between the user and the system.

By adopting this decomposition of the system design, it is clear that interaction design is a multidisciplinary topic and to design a usable system involves a range of heterogeneous knowledge from psychology to describe user needs and abilities, to graphical design to be able to design pleasant interfaces, to computer engineering to set up required technologies, both software and hardware ones.

The user interface design is important because we use computer systems in everyday life. Computers in early days were suitable only for primary tasks (e.g. programming, data processing). The user interfaces were not so important and often command line control was enough for experts that controlled the system. Nowadays computers are no longer intended for expert users but also for nonspecialists. Computers that are easy to use, easy to learn and allow to quickly complete the task are becoming more and more convenient for nonspecialist users. The importance of good user interface design can be measured by benefits that the design brings higher satisfaction of potential users. It also provides means to distinguish products from competing solutions. For business use, good user interface can lead to higher productivity of employees and their higher satisfaction. Bad designs can on the other hand reduce the productivity and there can bring higher risks of stressed employees. But showing benefits of good design can be difficult. It encourages us to ask how good design is actually defined and whether and how it can be measured.

A good interface design allows easy, natural and relatively fast-learning interaction between the user and the system. It provides means for users to quickly accomplish their tasks and clearly informs the user about errors in processing and/or about processing progress (e.g. deleting files, saving document). But this description of a good design has a disadvantage of being very subjective. Naturalness, fast-learning, quick task-completion – these are attitudes that are different for different people (or users). A metric of good interface design needs to be defined.

Beginning with the user part, knowing the set of users is quite an important thing. Before starting the discussion of a user part the terms *usability* and *user centered design* should be defined.

Usability is defined as a quality attribute that assesses how easy is to "use" particular user interface and it contains five quality parts [124]:

1. *Learnability* – How easy is it for users to accomplish basic tasks the first time they encounter new design?

2. *Efficiency* – How quickly can users that have learned the design perform specified tasks?

3. *Memorability* – Users do not use particular design for a period of time. When they return to the application, how quickly can they reestablish their skills?

4. *Errors* – Do users make errors during the interaction with a system? And are they able to cope with them?

5. *Satisfaction* – How much are users satisfied with the design?

The design process that supports development of easy to use interfaces is called *user centered design.* The user-centered design (UCD) is an approach to appliance design that enhances the design phase by using information about the people who will use the product [8]. The designers should involve users and user testing in all phases of the product evolution. There is also the international standard ISO 9241-210 *Human-centered design process* (formerly ISO 13407, withdrawn) that defines the typical set of steps for incorporating user-centered activities into the development process cycle. Main development according to this standard is formed by four activities. Fig 1.1 depicts the user interface development life cycle.

1. *Context of use* – Process of discovering the people that will use the system. What do they want to achieve during the usage and in which conditions will they work?

2. *Requirements* – Business or marketing requirements that need to be met for the product to be successful.

3. *Design solutions* – Propose design concepts from prototypes to complete product.

4. *Evaluate solutions* – Test the designed concepts, preferably with actual future users of the product. After this testing part, improve the design and evaluate it again.

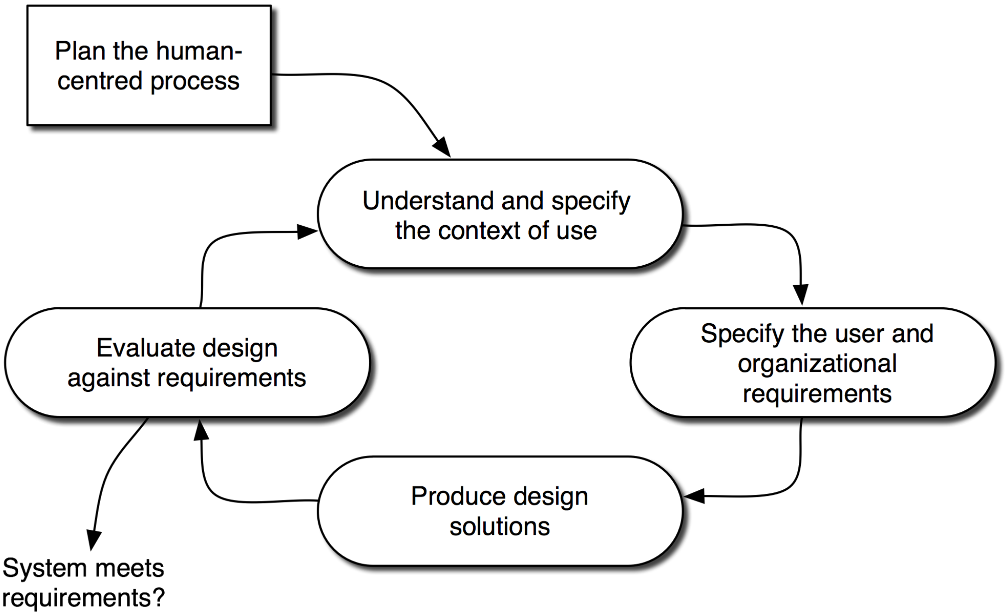

Plan the human-centred process
Understand and specify the context of use
Specify the user and organizational requirements
Produce design solutions
Evaluate design against requirements
System meets requirements?

Figure 1.1: User interface development life cycle (From ISO 9241)

Nielsen wrote that a three-day visit in one company to see user's working environment helped to find out over 130 usability issues [124]. So called *personas* should be used to model typical users in the beginning of design process. Personas are fictional personal characters that are created by the designers to represent various potential user types within targeted product domain. Designers often have a tangible idea of who their users will be. Knowing approximate age, abilities and other characteristics of their users, they are able to prepare personifications of these profiles. Typical personas with user profiles can help the whole development team to feel as they were in the role of typical user of the product [151]. Personas were introduced to human-computer interaction by Alan Cooper [31].

Interaction process can be defined as steps that users and an application do while the user is using the application. Users are only one part of the whole interaction process. The type of interaction is defined by the designed system and its use. The literature often puts a sharp distinction between input and output of the system; computer engineers often denote a printer as a passive output device and a keyboard as a simple input device. However, nearly all examples of human-computer interaction require both input and output to hand anything useful to the user. For example, most users would not appreciate a keyboard without the physical feedback of particular keys or without the feedback of written letter on the screen. The distinction between output devices and input devices becomes even narrower in the real world. A printed lesson book can serve as both output (printed text) and input (student's could make notes in the book) device [84].

Input and output devices permit computer (system) to interact with physical world. Human user interacts with computer through input/output devices. The simple ones are for example keyboard, mouse or display. But there are also advanced methods which assess the user state. They range from eye-tracking [120], facial expression recognition systems that are used to find out the current user mood [134], [25] to complex spoken multimodal dialog systems [205].

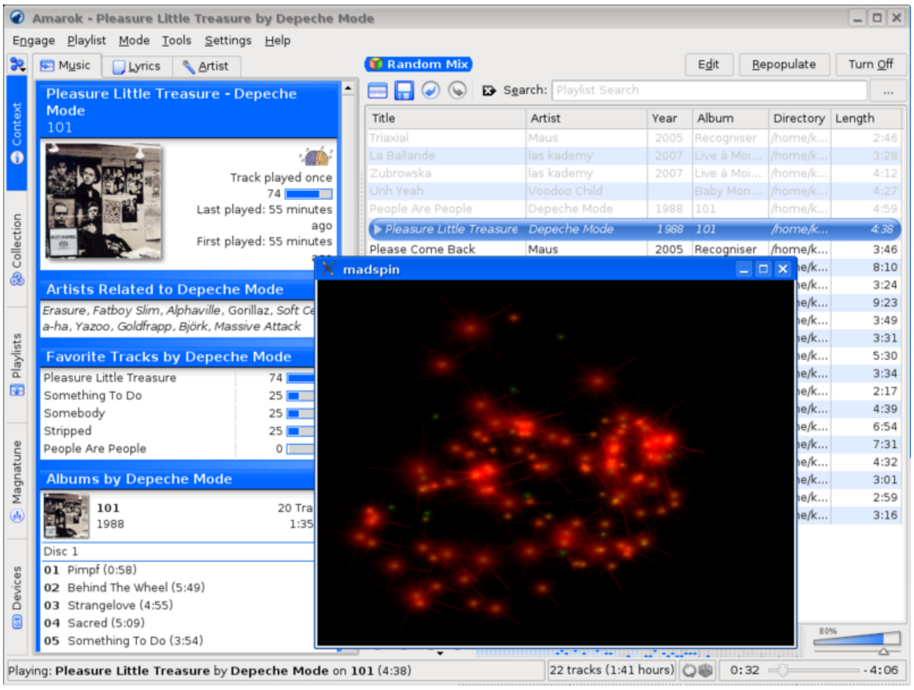
An interaction technique is a way of using a physical input/output device to perform a generic task in a human-computer dialog [55]. In the "classical system interfaces" users are pushed to learn new procedures to accomplish a particular task. Further in the text the term *classical user interfaces* will be used for keyboard and mouse based graphical user interfaces (see Fig. 1.2). Nowadays the role of designers is going to turn reverse. They start to develop various "reality based" interaction techniques. These new interfaces are trying to exploit human pre-existing knowledge and pre-computing skills[1]. Imagine comparison of a person operating a system using graphical interface and another using a virtual reality walk through. The graphical interfaces obviously involve learning how to reach the task target. While the user can utilize previously learned skills to navigate the system in the virtual reality walk through. The new reality based interfaces have some commonalities [83]:.

- *Embodiment*, human-like characters rendered by 3D graphics, expressing emotions, using speech, etc.
- *Interaction in real world*, e.g. augmented reality – reality view that is augmented by computer generated user interface
- *Full body interaction*, sensing user's body movements, emotions, recognizing speech, etc.
- *Doing other things than the main task*, interfaces with low cognitive load to help users to cope with information overload
- *Mobile interfaces*, improving mobile device accessibility considering small screen estate
- and others …

Accessibility

When taking into account specific (or even social) groups of users as for example older adults, there are several problems or limitations for using some interaction techniques. Classical interfaces (see Fig. 1.2) are not very suitable for users with some disabilities. Background of this suitability is a degree of adoption of such interfaces and technologies. There were several studies of computer user interfaces adoption by older adults. Lot of older people feel less comfortable and less self-confident than younger adults while working with computers [122]. Dyck et. al. showed that older people with previous experience are more comfortable to use computers than people without any experience [45]. Targeting these specific groups of users imposes further requirements on the skills of involved designers (e.g. creating new suitable interaction techniques). The needs of older people or people with some disabilities are examined by the field of accessibility. *Accessibility* investigates the problem of making products, software, etc. accessible to various groups of people (including handicapped people).

[1] Skills that are not achieved obtained by working with a computer, e.g. paging through newspaper



Pleasure Little Treasure - Depeche Mode
101

Track played once
74
Last played: 55 minutes ago
First played: 55 minutes ago

Artists Related to Depeche Mode

Erasure, Fatboy Slim, Alphaville, Gorillaz, Soft Ce
a-ha, Yazoo, Goldfrapp, Björk, Massive Attack

Favorite Tracks by Depeche Mode

Pleasure Little Treasure	74
Something To Do	25
Somebody	25
Stripped	25
People Are People	0

Albums by Depeche Mode

101
1988
20 Tra
1:35

Disc 1

01 Pimpf (0:58)
02 Behind The Wheel (5:49)
03 Strangelove (4:55)
04 Sacred (5:09)
05 Something To Do (3:54)

Title	Artist	Year	Album	Directory	Length
Triaxial	Maus	2005	Recogniser	/home/k...	2:46
La Ballande	las kademy	2007	Live à Moi...	/home/k...	3:28
Zubrowska	las kademy	2007	Live à Moi...	/home/k...	4:12
Unh Yeah	Voodoo Child		Baby Mon...	/home/k...	4:27
People Are People	Depeche Mode	1988	101	/home/k...	4:59
Pleasure Little Treasure	Depeche Mode	1988	101	/home/k...	4:38
Please Come Back	Maus	2005	Recogniser	/home/k...	3:46
				he/k...	8:10
				he/k...	3:24
				he/k...	9:23
				he/k...	3:49
				he/k...	3:31
				he/k...	5:30
				he/k...	3:34
				he/k...	2:17
				he/k...	4:39
				he/k...	6:54
				he/k...	7:31
				he/k...	4:32
				he/k...	3:01
				he/k...	2:59
				he/k...	3:16

Playing: **Pleasure Little Treasure** by **Depeche Mode** on **101** (4:38) 22 tracks (1:41 hours) 0:32 -4:06

Figure 1.2: Amarok music player interface. Example of classical mouse and keyboard based interfaces. (Source [93])

1.1 Problem Definition

Users want to control UIs without long and tedious learning path. They want to interact naturally. Speech is one mean that can be used to develop easy user interfaces. The ultimate goal of human-computer interaction designers is the implementation of systems that are able to interact with humans in a natural conversational way.

A voice based user interface is one of the possible human-computer interaction methods. Last few years have brought many advances in automatic speech recognition systems (ASR), natural language understanding (NLU), dialog management (DM) systems, and in text-to-speech systems (TTS). These systems are getting more and more sophisticated and as such they have grown relatively complex. The quality of each system is also dependent on other systems. For example ASR recognition error rate can be partly determined by the quality of dialog manager that can define proper recognition grammars in a specific dialog context. The systems and their connection are depicted in Figure 2.2 and described in Section 2.1. A good example of the speech applications trend could be automatic telephone systems which help users solve simple problems and, in difficult cases, reroute the customer to appropriate human operator. The AT&T How may I help you? system is credited to be the first among such advanced systems [65]. However, computer spoken dialog technology is still far from having a near-human performance [143].

A spoken dialog system should present some benefits over other styles of interaction. Spoken dialog systems are the most suitable when they enable something that cannot otherwise be done (e.g. safely operate a phone or a navigation system while driving); or when the user's hands and eyes are busy; or when the keyboard is not available. The usage of spoken dialog systems could overcome some usability issues for specific groups of people: for example, tremor and age-related changes in bodily motor control represent a problem for navigating touch user interfaces; or people with different visual disabilities can hardly use visually based interfaces. Such groups of people could clearly benefit from voice-based interaction. Non-disabled users often prefer a menu and a "common" based style of interaction[2] to spoken dialog systems, e.g. in situations when the user is not in a private environment [98] or in noisy environment (speech recognition errors).

1.1.1 User's anxiety of talking to computers

As the previous section stated spoken dialog system can increase the naturalness of interaction and improves the learning curve for users. Despite these positive improvements the voice-based interfaces have one uneasy feature. People are used to that just humans can speak but talking to machines is not considered as common. Users do not like to speak to black boxes and machines. The communication with machines seems to be weird. However, this slowly changes in the era of various digital mobile assistants that are voice-enabled, e.g. Apple Siri [7], Google Now [64]. These systems allow to control personal information as calendar, e-mail, etc. by natural language commands. They even allow to search information in the Internet.

The second part of the user's anxiety issue is well visible in special groups of users. Often elderly or people with some kind of psychical illness prefer communication with humans. They need the feeling of personality to express their user goals.

One possible solution to this problem can be providing some substitution of human personality in a form of human-like agent rendered as a part of UI. The human-like agent, sometimes called Embodied Conversational Agent (ECA) or interface agent is an agent that interacts with surrounding environment through the body (or parts of body, e.g. face, head, hands). The agents are represented graphically as humans, cartoon characters or animals. Specifically, agent is understood as an autonomous entity that acts based on information from environment to achieve a goal, e.g. convey message. From now on, we will assume humanized user interfaces as applications that employ human-like talking agent to convey information to the user.

Humanized UI agents are applicable in various areas where a user is intended to practice (learn) or use speech. For instance, these include soft-skills education software or areas where one can not use mouse/keyboard style of interaction, e.g. immersive virtual reality systems, augmented reality, virtual tour guides and others. These humanized UIs are not very useful in areas where direct manipulation is needed, e.g. office systems, painting/creative applications. These sorts of applications are much better managed by keyboard/mouse/tablet interfaces. Humanized UIs are not suitable in environments where user's visual sight is needed to perform important tasks (e.g. driving cars, controlling industry machines).

In the human-interaction area there exist debate over the topic whether direct manipulation is better than interface agents [169]. One group prefers direct manipulation. They highlight the fast response times such attitude and unutilized visual potential of human, even 40.000 icons on one desktop is not a problem for human user.

The other group highlights that interface agents can bring proactivity to the human-computer interaction. Interface agent can take initiative because it inherently knows the

[2] The common style of interaction means keyboard, mouse, touch based systems

current context to be able to communicate with user. It can also adapt to changing user's interests. Interface agents make computers available to naïve users.

Nass et. al. have shown in their experiments that experienced computer users apply social rules to the interaction with computers, even without human-like agents [123]. This can be true but without human-like agent one is not able to convey e.g. facial expressions.

Interface agents belong to the area of mixed-initiative interfaces, i.e. interfaces where one assumes that users and the computer program may collaborate to achieve the user's goals. The main principles of such interfaces are summarized in the work of Horvitz [79].

The key principles for our work are:

- Considering uncertainty about a user's goal. In the light of uncertainty of all speech-based systems, the system cannot be sure about user's goal.

- Considering the status of a user's attention in the timing of services. The agent should consider timing of actions and presentation according to the user context.

- Employing dialog to resolve key uncertainties. If the system is uncertain about the user's action it should employ dialog with user to clarify these goals.

- Employing socially appropriate behavior for agent-user interaction. The agent should be human-like, e.g. turn-taking during a dialog.

There exist domains where humanized UI can be really valuable, e.g. the area of serious games. Serious games have become a big topic and interest in the development and research of such games is growing. They can be defined as games to educate, motivate and change behavior of users of different ages. Video games industry succeeded in providing immersive reality games using new technologies like high resolution 3D video and audio, social collaboration in games and input based on sensors. Both genders and all age groups are attracted to gaming. The development is driven by entertainment purposes only. On the other hand, serious games are building on this entertainment value but they usually add value through educational potential, e.g. to teach kids basic math or animal species [156].

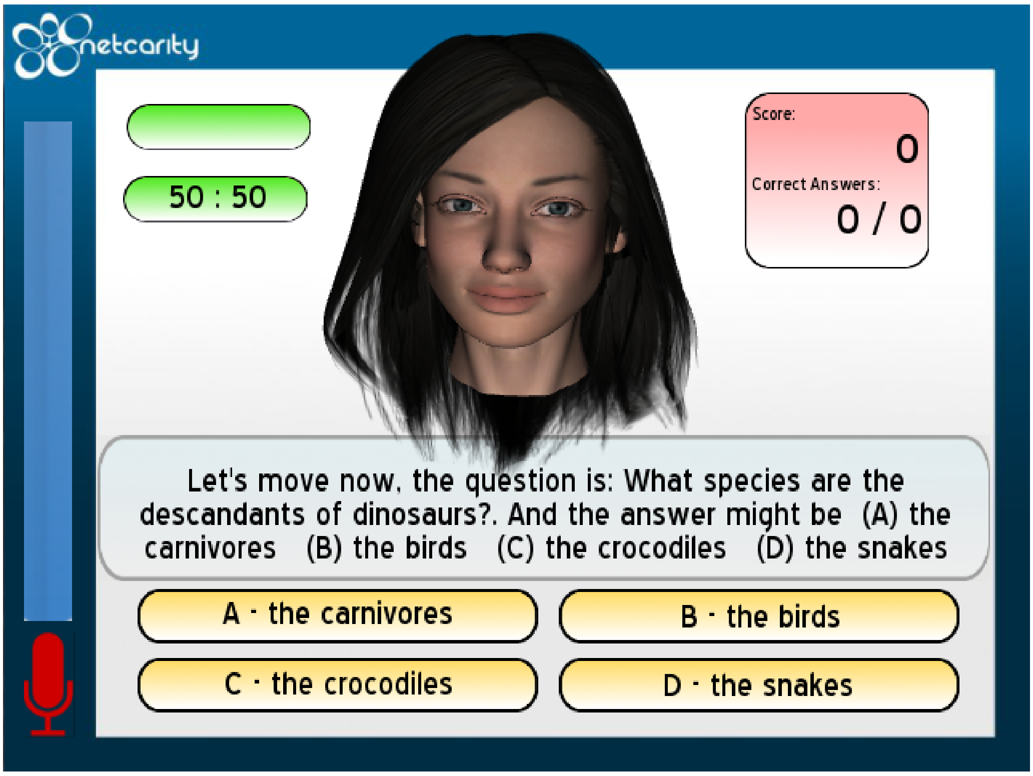
In serious games, gaming element should be prevalent. The enjoyment depends on the user, environment and situation. Some prefer competition as the most enjoyable factor while others find their way in role playing, creative work or some moderately challenging activities. To distinguish serious games from games meant for entertainment only, serious games are defined as an interactive-based computer software for one or multiple players that was designed to be more than entertaining.

1.1.2 List of ECA Applications

This section lists example of various ECA applications. We implemented these application to drive the development of whole complex ECA environment. The first one introduces a serious game called Billionaire.

Billionaire ECA Game

World of games is sometimes seen as the world of children but games can be dedicated to various age-groups, i.e. older adults. The serious game called "Billionaire" is used as an example of humanized user interface. The concept of the game is borrowed from a famous TV game "Who Wants to Be a Millionaire"? This game should be entertaining in the way of overall interactivity using humanized user interface with a talking head. The talking head works here as a moderator and talks to the user and user answers back to the talking head. See Figure 1.3 for UI screen shot. The game have been developed as a part of NETCARITY EU project[3] and it is further used as a main platform to show solution to problems defined in this dissertation thesis.



Let's move now, the question is: What species are the descandants of dinosaurs?. And the answer might be (A) the carnivores (B) the birds (C) the crocodiles (D) the snakes

Figure 1.3: Screenshot of Billionaire game with humanized user interface

Netcarity was a successful EU project exploring, testing and improving technologies that will help older adults to improve their life, day-to-day needs, independence and safety at home. The project tried to increase quality of life by reducing their social isolation and bringing activity possibilities. This was done through promotion of user-centered design of solution implementation. Development was focused on new interface modalities which are suitable for older adults and to integrate the systems to one. The pioneering Netcarity System for ambient assisted living (AAL) has been installed in 84 trial homes and it is being tested by real world users.

[3] http://www.netcarity.org

Who are older users?

The ageing process is of course a biological reality which has its own dynamic, beyond human control. The age of 60 or 65 is said to be the beginning of old age. In developing countries, old age is seen to begin at the point when active contribution is no longer possible [66].

In the old age often following limitations arise during the normal ageing process [6]:

- Visual decline – decreasing ability to focus on near tasks, color perception, contrast sensitivity, reduction in visual field
- Hearing loss – inability to hear high-pitched sounds
- Motor skill issues – tremor (trembling in the hands), rigidity, postural instability
- Cognitive decline – loss of memory, confusion, mood changes, communication problems
- Multiple sensory loss – combinations of limitations

One possibility to bring new activities to older people is to create everyday morning habit. The Netcarity introduces good-morning service scenario that comprises of phone calls from service agency to check the health and overall condition of older person. Part of this scenario can be applications that can measure the conditions automatically through playing a game which can be interesting to the users, e.g. "Billionaire game".

ECA-based MP3 Jukebox

We developed a spoken based dialog jukebox application that uses ECA as primary interface. The jukebox allows to play music from the user's computer. This development was done in order to debate practical issues and investigate new potential for personification of user interface and customization of ECAs.

The graphical interface is represented by the ECA and a visual information (including audio, video, pictures) is displayed on the background (see Figure 1.8). We explored the feasibility of this approach by understanding the key points of user acceptance from the viewpoint of multimodal conversational systems. The prototype has been subjected to usability testing in order to obtain qualified and quantified evidence and understand potential pros and cons of implemented ECA interface.

The usability study confirmed the added value of ECA-based jukebox for entertainment conversational applications. Detailed description of ECA-based jukebox application can be found in [A.9].

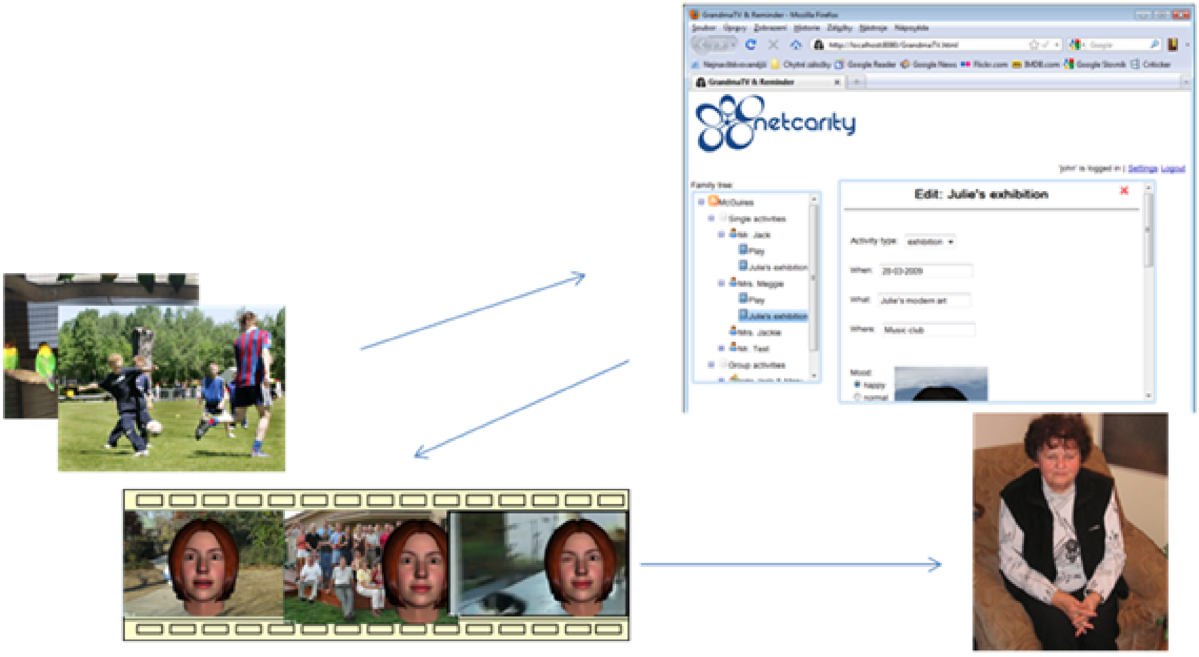
GrandmaTV Concept

GrandmaTV is an experimental presentation system that can generate dynamic ECA-based presentations from structured data including text context, images, music, sounds and videos. Thus, the ECA acts as a moderator in the chosen presentation context, typically personal diaries. The GrandmaTV was an experimental concept which used the presentation engine on a social inclusion service in the Netcarity project.

The presentation engine can turn individual event entries (or any relevant structured data) into a multimodal presentation with talking head as a moderator. ECAs represent a rich channel for conveying both verbal and non-verbal messages in human-computer interaction. Hence, we could utilize this human-like capability to convert “dry” data such as diaries and blogs into more lively and dynamic presentations. This is achieved by transforming the individual blog items such as text, messages, images, music and video into a multimodal stream running on the background and the ECA acting on the foreground to convey these messages and connect individual information pieces together through narration.

The storylines of specific narrations are generated from multiple scenario templates. These prescriptions are generic and prepared manually by expert users. They define content that can be presented by GrandmaTV system, e.g. utterances as building blocks. For example, each member of family (or family group) has parametrized textual template that describes his/her activities. The template text is also enriched by expression, head position or various modal metadata that help drive the ECA behavior.

Personal input data are stored in form of activity feed sorted by time. One activity is represented by tuple of parameters, e.g. activity name, activity type, person, date/time, textual information, photos, videos, etc. Figure 1.4 shows the GrandmaTV input/output data life-cycle. The user (grandchild, child, etc.) fills in activity data into the web application; these data are processed by our GrandmaTV framework and converted into ECA anchored video which is then viewed by grandparents. Further information about GrandmaTV concept can be found in [A.4].



Edit: Julie's exhibition

Activity type: exhibition

When: 20-03-2009

What: Julie's modern art

Where: Music club

Mood: ● happy ○ normal

Figure 1.4: GrandmaTV Lifecycle

ECA-based Multimodal Kiosk

Previously mentioned ECA-based systems were targeted mainly for personal use. The multimodal kiosk represents system that serves as entertainment and information source in the public space, e.g. offices, school, institutions, receptions. People can interact with ECA and ask for information (e.g. weather, lunch menu) or play various interactive games (e.g. math game). The setup is depicted in Figure 1.5.

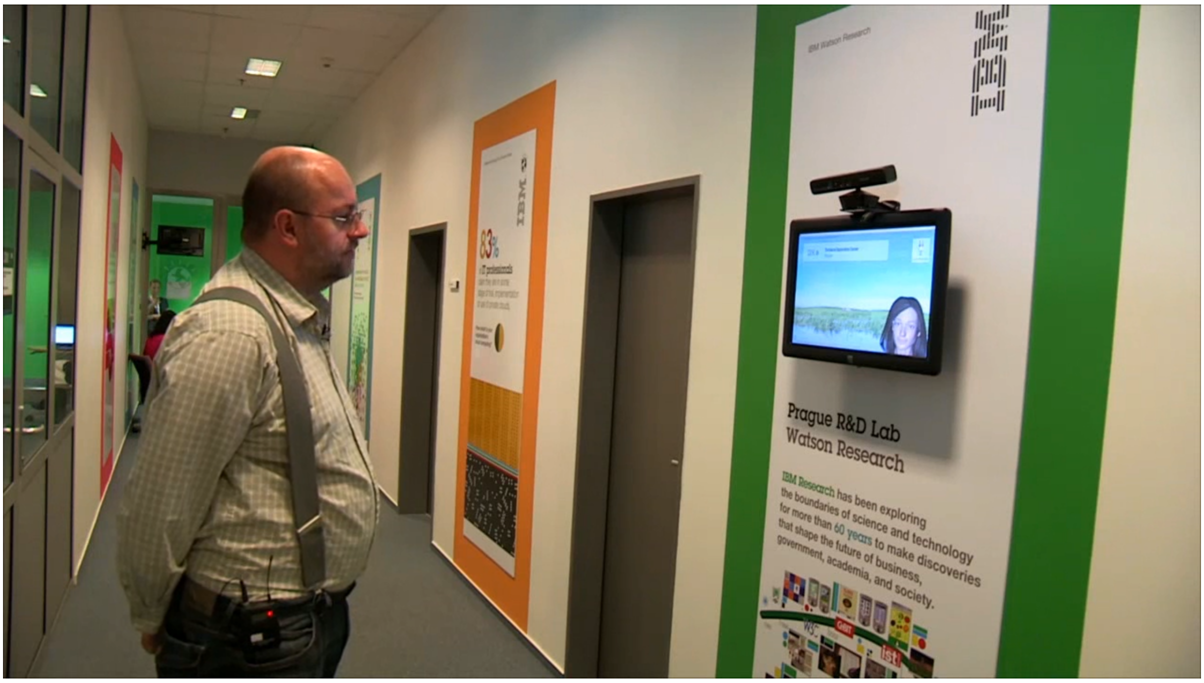

IBM
Prague R&D Lab
Watson Research
IBM Research has been exploring the boundaries of science and technology for more than 60 years to make discoveries that shape the future of business, government, academia, and society.

Figure 1.5: User interacting with multimodal kiosk

The kiosk is composed of a display which shows a talking head and multiple visual background information. The kiosk application accepts input from users multimodally through voice and/or gestures captured by the Microsoft Kinect device. Actual first usage of this multimodal kiosk was to act as virtual router for people passing by. The kiosk is situated close to lecture rooms and it can show users what lecture is in which room and give them directions to the rooms.

This first installment was further extended by multiple functionalities, e.g. weather forecast, joke telling, smalltalk). Even a little interactive games were implemented in the current versions of the multimodal kiosk, e.g. math game (see Figure 1.6).

The ECA-based kiosk platform have become platform to test various multimodal technologies (e.g. speech, gestures, ECA) working in a synergic way.

1.1.3 Naturalness of a Dialog with Talking Agent

One of the main challenges when using humanized UI is trying to make the head or ECA act in a very natural way as humans do. The naturalness of interaction between humans has many aspects. Measuring the naturalness of interaction human-computer represents complex question.

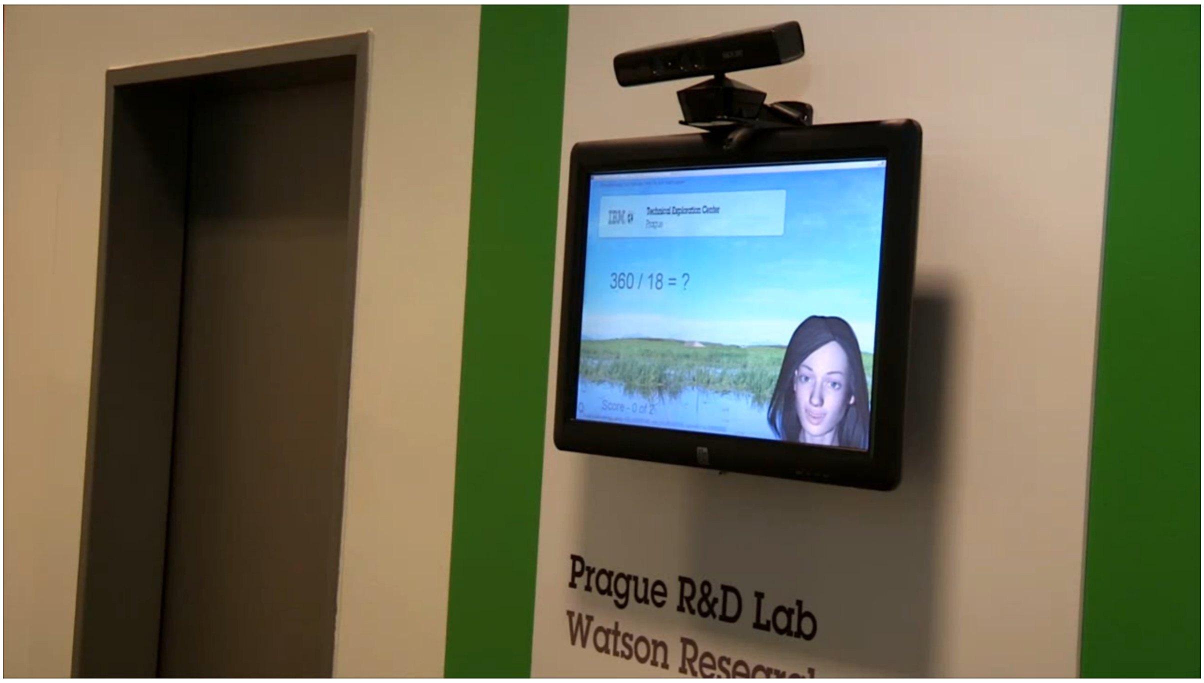


Figure 1.6: ECA-based multimodal kiosk – detail

Measuring Naturalness of Talking Agent

Many researchers try to measure the naturalness of close encounters with their implemented ECAs. Subjective evaluations are not real solution. The questionnaires are not comparable to other studies due to controlling all the potential environmental variables. Objective metric is needed to compare different ECA designs. There exist three types of metric:

1. Psychological metrics based on psychological tests

2. Behavioral metrics based on comparing the behavior of participants to some standard behavior

3. Physiological metrics based on measuring the internal state of human participants

There are many physiological signals related to the human internal state, e.g. Galvanic Skin Response, Blood Volume Pulse, Respiration Rate, Skin Temperature. The research was done to find out usefulness of these signals [118].

However, improving the naturalness of ECA interaction can lead to negative results. Roboticist Masahiro Mori found out associations of so called uncanny valley with robot design. The more human-like a robot became, the more familiar and likeable was to a viewer, until a certain point was reached. Then the robot became rather strange than familiar [107]. The situation is shown in Figure 1.7.

One important naturalness aspect is believed to be how the system/user yields or takes turns in a spoken dialog.

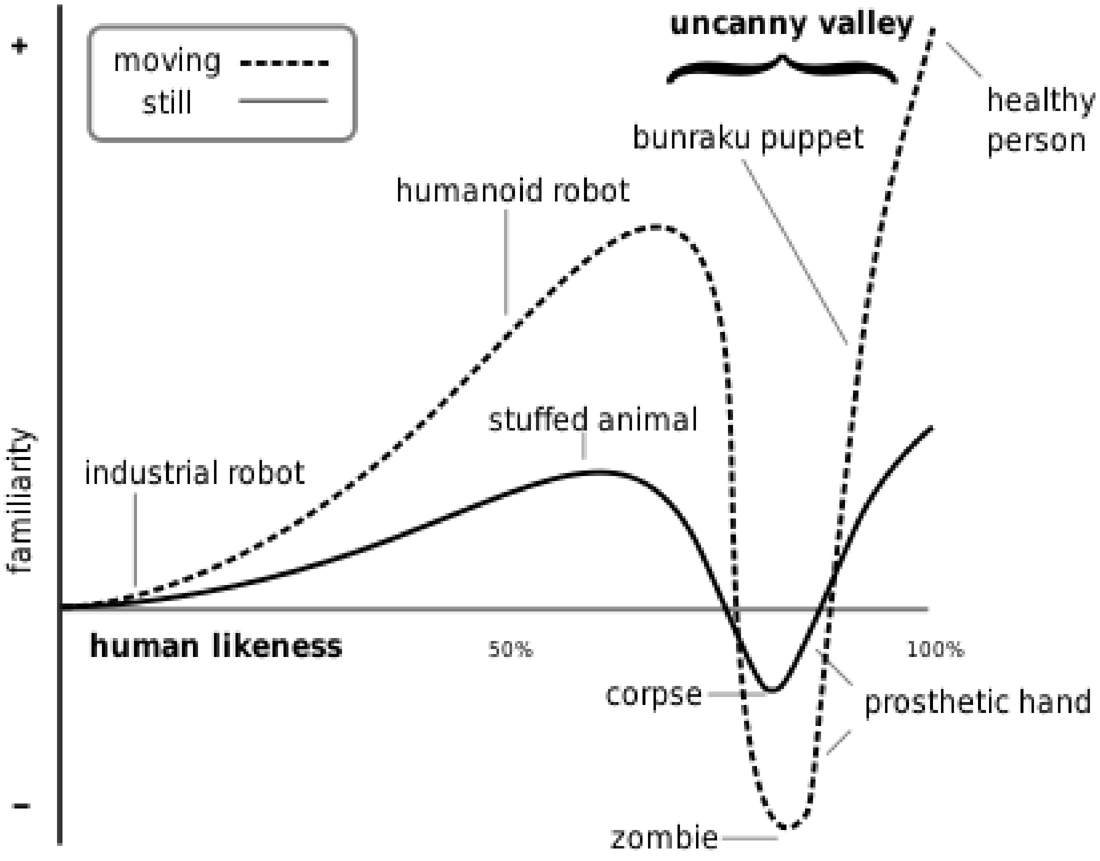


Figure 1.7: Perception of ECA/robot naturalness – uncanny valley (from [107])

Issue: Dialog Turn-taking and Turn-yielding

A human sends plenty of cues during a natural conversation to indicate a wish for turn-taking or turn-yielding. However, spoken dialog systems use predominantly only one way to yield the turn nowadays, and that is the use of a long pause [67], typically in the range of 0.5s to 1s [54]. But long pauses are not so natural in human-to-human dialog; conversations in general tend to be smoother without them. Duncan analyzed face to face conversations in English. Together with Fiske, he also stated that speakers display complex signals at turn-endings [44]. Most of these analyzed signals were spoken (audio) ones.

It should be taken into account that verbal (speech) communication is not the only part of interpersonal communication. The other part is nonverbal language (facial expressions, body gestures, etc.). While listening to others, people do not focus only on the verbal content of a conveyed message. During complex assessment of a speaking person we process both parts of speech: the nonverbal and the verbal [70].

Incorporating some form of face-to-face communication into spoken dialog technology could enable the system to express nonverbal parts of communication. Seeing virtual faces (that means conversational agents, avatars, etc.) also humanizes computer user interfaces and makes them more acceptable for common users [203]. These so called

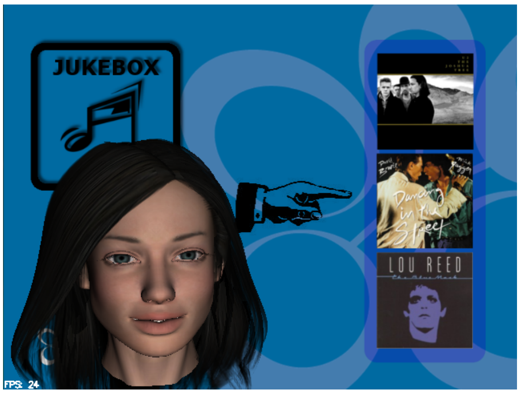


Figure 1.8: Example of proposed Talking head interface for MP3 player. The head is navigating the user through his/her music album collection. Examples of some ECA's gestures (head movement, background picture, pointing on the screen)

embodied conversational agents (ECAs) or avatars, integrate gestures, facial expressions and visual / nonverbal aspects of speech into human-computer interaction [23]. ECAs also allow researchers even more to analyze multimodal turn-taking and turn-yielding cues. Using a richer range of turn-yielding cues for talking agent should bring more natural dialog for an ECA system.

Current spoken dialog systems prevalently use push-to-talk button to indicate start of the speech recognition. This approach is not very natural for human users. It should be better to extend push-to-talk system by employing visual or vocal turn-yielding cues on ECA.

Issue: Combination of Classical GUI Interfaces and Humanized Talking Agents

The crucial part of implementing ECA based user interface is the synchronization of output communication channels (modalities). To make the whole ECA application consistent there should be a system that generates and synchronizes all the output modalities. Without synchronization the user of the system could be confused by an inconsistent mix of several output modalities. There is an authoring language on the low-level layer that helps developers to drive the behavior of an agent and handles synchronization. There is plenty of such languages, e.g. MPML [202], SMIL [128], etc.

These low-level operational languages have one common issue. The means to express and use knowledge that would be helpful to automatically generate gestures and behavior of embodied conversational agent do not exist (e.g. face expressions, showing background images, playing video, pointing in the screen, moving head). The authoring language should allow for seamless combination of classical GUI interfaces with humanized talking agents. The example of such a combination is shown in Figure 1.8.

We propose better authoring language for ECA-based applications. This language needs to be designed with the user-centered view in mind. Users are typically developers of ECA applications. Most of the features should be designed in response to the needs of developers. Structural composition of the authoring language should follow these requirements:

- fast learning curve
- designed to be useful for real-time controlling of ECA interface
- extensible for future ECA functionalities
- ability to control both ECA and the surrounding user interface at the same time
- based on the principle of mixing various interaction channels (e.g. eyes, head position, facial expressions, multimedia, background)

While the most of the development effort in the field of ECA applications is currently spent on improving the graphical appearance of the ECAs, e.g. skin rendering [41], there has been little done towards the need of investigating the interaction models that will supply the interaction logic and behavior patterns for these human embodiments. The interaction logic of humanized interfaces is predominantly solved by opening up low-level controls of ECA to developers. However, this relays most problems with ECA-user interaction to application developers.

We aim exploring the development of ECAs from this new perspective. The ECA platform should be easy to use for developers of spoken dialog systems. These voice and GUI only systems have potential to be easily converted to use an ECA as a new interaction channel. In the beginning of these conversion efforts it is good to shield developers from complex authoring of ECA behavior. Providing only low-level authoring means leads to large demands on the dialog manager output. The dialog manager needs then to precisely control the behavior of ECA. Allowing to use some default behavior patterns can speed-up the process of spoken dialog system conversion and make it straightforward. However, it is important to leave the freedom of changing the ECA (interface) behavior tuning as rich as possible. During the application development some behaviors can be modified, improved or turned off.

We propose an ECA that uses ontological modeling to drive its interaction capabilities. Such an ECA's ontological knowledge base will drive the interaction logic and supply proper behavioral patterns, including the voluntary gestures (e.g. specific facial accents when a sentence starts/ends, speech), the involuntary head gestures (e.g. automatic head movement), eye and head position during interaction, facial expressions and eye blinking frequency. These interaction patterns should be generated in particular context of application state and user behavior. This will simplify the development of ECA-based applications as the proper behavior patterns are selected based on the context. Thus, application designers can focus on content of conveyed messages.

We build some form of "behavior & UI architecture" for an ECA, that allows to experiment with ECA interfaces. Developer can use default behavioral patterns or plug new patterns based on some experiments or assumptions about human behavior and run usability experiments to evaluate these behavioral patterns. The principle of such architecture would be similar to various programmable cognitive architectures simulating human mind, e.g. ACT-R [3], SPEAR [99].

Issue: Improvement of ECA Evaluation Methodology

Development of ECA-based applications or applications with user interface comprises the usability testing during the whole process. Multimodal systems trying to mimic human-like behavior need relatively complex testing because of rich communication channels. There are several aspects that needs to be tested with human user in ECA applications. Important ones are summarized by the following items:

- *ECA appearance* – How pleasant are the behavior parameters to a user? These parameters can be static or dynamic.
- *Turn-taking behavior* – How an ECA takes or yields a turn back to a user (including backchannels)?
- *Visual articulation* – Quality of speech visualization.
- *Speech comprehension* – Quality of speech recognition (e.g. measuring Word Error Rate – WER) and quality of dialog management. Detailed description is in Section 2.1.
- *Usability testing of an ECA application* – Complex testing of spoken dialog application with ECA interface

Section 5 presents several evaluation studies where evaluation design and metrics are defined. The evaluation studies cover areas mentioned in the previous list. Some of evaluation studies try to further improve the experiment design, especially in the case of visual articulation test.

The McGurk effect test is a method used to evaluate speech articulation of talking agents. The McGurk effect is based on presumption that humans use both hearing and vision senses in parallel to perceiving speech [115]. This can be verified by simple experiment where the participants are presented with videotape of visual *ga* syllable with audio *ba* syllable. A great deal of participant should perceive that the *da* syllable was pronounced.

The McGurk effect is used as one of the articulation evaluation test to asses quality of ECA's articulation [110] and [33]. Participants of this experiment are given synthetic ECA McGurk sequences and their confusion responses are measured. The participants of such experiments are humans with normal hearing and vision.

One issue with this experiment is that authors do not mention whether participants had corrected vision or not. As the quality of sight can influence the visual perception of speech, this can be an important factor in these measurements.

This thesis should find out whether there are differences in the perception of McGurk effect among normal vision participants and participants with corrected vision.

Human-robot Interaction

Software talking agent is one form of human-like presence in the virtual world. Another form of talking agent could be more real than virtual, robots. There is an emerging field of research that deals with issues of human-robot interaction. Robots differ from the software based talking agents in that they are often designed to be autonomous and mobile. They have increasing potential in the fields of health care or as companions for older adults [167].

Robots involve human-inspired behavior. During spoken conversations between humans and robot it is important to maintain goal-oriented dialog with alignment and accurate timing of gaze, gestures and posture signals. The dialog should be constructive. Rational agents coordinate their actions and perceive the environment. They react to the situations in line with their beliefs, intentions and understanding. They maintain shared knowledge to maintain social bonds [86]. The robots use multimodal signaling, e.g. gesturing, eye gaze, changing posture, turn-taking always in synchrony with another actions.

Robot should be capable of social communication and behavior when interacting with users. Fong et. al. defined several characteristics of social behavior [56]:

1. express and/or perceive emotions
2. communicate with high-level dialog
3. learn and recognize models of other agents
4. establish and maintain social relationship
5. use natural cues (e.g. gaze, gestures)
6. exhibit distinctive personality and character
7. learn and develop social competencies

The solution proposed in this thesis can be potentially extended to this relatively new research field. The turn-taking and turn-yielding schemes are important factor in the fluency of communication between people. People often share the resources in the common place of interaction among each other. The conversational floor is one of such shared resource, others include e.g. borrowing the pen, sharing drinks and food. The action of robots when interacting with humans should be interruptible as much as possible. One turn-taking scheme for robots is examined in [24]. They used Petri net representation to create system for the control of turn-taking interactions flow. This led to a time reduction of task completion.

1.2 Outline of Dissertation Thesis

The dissertation thesis is organized as follows. Section 1 provided introduction to the problems solved by this dissertation thesis. Section 2 offers state of the art survey of speech-enabled applications and humanized user interfaces. Several techniques needed for building speech-enabled applications with humanized interfaces are discussed. Section 3 describes solution proposals to the issues in humanized interfaces. Section 4 outlines our solution for selected issues. Section 5 shows the evaluation process, describes conducted experiments and discusses results. Section 6 summarizes the thesis, provides advantages and disadvantages of our solution and presents possible future work.

2 State of the Art

The research of humanized user interfaces is a very broad and multidisciplinary field. It comprises of many domains like psychology, computer graphics, human-computer interaction, speech recognition, speech generation and dialog management. This chapter describes the research needed to solve issues sketched in Section 1. Firstly, structure of spoken dialog system is reviewed, mainly dialog management and turn-taking. Second part of this chapter is dedicated to the Embodied Conversational Agents, including ontologies for their knowledge bases.

One of many possibilities to improve classical interaction is to use voice-based interfaces. Properly working voice-based interfaces are the main goal of speech community. The ultimate goal is the implementation of machines able to interact with humans in a natural conversational way to provide services that would otherwise require human operators or menu (graphical) based systems [144].

First implementation of program that employed natural communication between man and a machine was ELIZA [191]. This was a simple chat-based system that employed pattern recognition to reply to the user's input sentences.

Although plenty of time passed from the first commercially known spoken dialog system from AT&T "How may I help you?" [65], the performance of computer spoken language technology is today still far from near-human performance [142]. But there are systems that assume these limitations (e.g. speech recognition errors and/or multiple speech recognition hypotheses) and provide the needed freedom of expressivity for user, for example Dialog Strategies for Call Routing [196].

2.1 Spoken Dialog Systems

Human speech is one of possible interaction methods. This method has some advantages over other/"classical" interactions. An overview of the current state of the art in the field of speech-enabled applications is provided here. Speech is the main communication channel for humans, so no special training is needed to use speech interaction. Speech also leaves user's hands and eyes free so it enables humans to interact with the system in parallel with some other activity (e.g. driving, cooking, operating industry machines). Another advantage of speech is that it can be transfered relatively cheap and fast (e.g. phone lines), microphones are also smaller devices compared to keyboards.

However, speech also brings some disadvantages. Humans do not like to talk (transfer information) in public spaces. This paradigm is even stronger when the information is private (e.g. giving a credit card PIN code). Speech is serial – the information cannot be transferred in parallel, one must wait for the whole utterance to get grasp of the transferred view unlike in a case of visual sense. It is slow when describing e.g. location or presenting long lists of options.

There are several areas where the usage of speech is ideal and sometimes even the preferred way of interacting with computer systems [91]:

- *Navigation in complex environments* – Complex environments represent e.g. nested menus or tool bars full of software functions. It is better to use speech for interaction in applications that can do many tasks known to the user and their controls would be otherwise hidden in complex and wide menus. Speech can be possibly combined with other modalities [29]

- *Navigation in long list* – Imagine very long contact list where there are names of your friends in a mobile phone. Having to search this contact list is faster and more comfortable using your voice than using touch or keyboard controls, if one knows the name [85]. However, searching by the name substring can be comparable fast.

- *Naïve users* – If one wants to provide support of complicated environments for naive users, speech seems to be the right choice. Interacting with the environment using voice is much more natural than "classical" UI means (e.g. train timetable services – German timetable system [47] and RAILTEL [14]).

- *Low bandwidth needed* – Transferring information by speech requires low bandwidth. Thus, speech can be a modality suitable for use in areas where e.g. only phone line is available.

- *Minimal distraction* – Speech provides means of interaction that is distracting hearing senses but visual and haptic senses are free for other activities. This predetermines speech for the environments where low visual distraction is needed, e.g. car [97], [175].

- *Handicapped people* – For some handicapped people speech is the only possibility to interact with systems.

Since the early 1990s, many spoken dialog systems were developed both in the commercial and in the academia domains. Early systems represented simple call routing systems like "How may I help you?" [65] or travel planning [187]. Recently developed systems are used in very broad environments, for example in car applications "short messages dictation systems" [34] or navigation systems [192]. Specifically, the EU project GetHomeSafe is focused on development of systems that will be used in car navigation, entertainment and car-communication [30]. Last but not least several multi-domain focused systems reached wide group of users in a form of spoken dialog systems as mobile assistants, such as Apple Siri [7], Google Now [64] and Nuance Dragon Mobile Assistant [130]. These systems support multiple user tasks and access information from various sources and web services that are available on the Internet.

This spread of mobile speech assistants was facilitated mainly by improvements in the area of speech recognition. The performance of various speech recognition engines is measured during Automatic Speech Recognition Evaluations at NIST [126]. Figure 2.1 shows history of evaluations results. The performance of speech recognition is often measured by the Word Error Rate (WER) metric. The WER is derived from the Levenshtein distance, on the word level [103]. Overall the performance of large vocabulary speech recognition in quiet conditions improved from 90% WER to 10% WER now. However, there is a long way to get the speech recognition to the level of human error, especially in noisy environments.

Commercial or research prototypes of dialog applications use similar application architecture that consists of voice recognition system, natural language understanding module, dialog manager and text-to-speech engine (see Fig. 2.2).

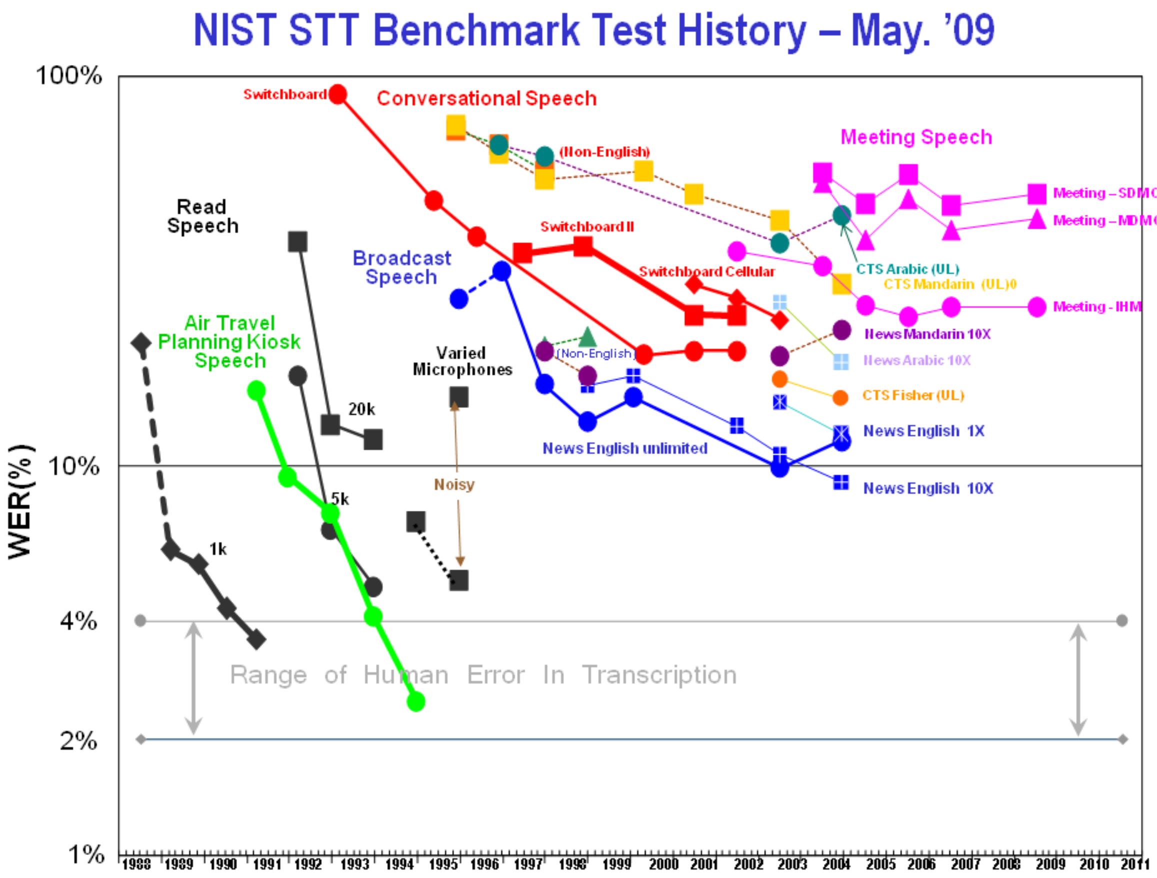


Figure 2.1: Automatic speech recognition performance improvement in timeline (from [126])

- *User input* – user usually speaks to system's microphone and speech signal along with noises is the input to the whole system. However, there exist dialog systems that allow typing questions using keyboard.

- *Automatic speech recognition* – first this module transforms speech signal to a sequence of parameters. Secondly, speech recognition algorithms transform these parameters to a sequence of words/sentences.

- *Natural language understanding* – this part of system allows to extract "meaningful information" from user's utterance. Mostly this is done by morphological analysis, part-of-speech tagging and named entity extraction. This module converts the recognized utterance to user's goal/intent and provides slot values for extracted named entities. For example if the user's utterance contains a city (e.g. New York), the slot CITY will be filled with "New York" words. This extraction is done mainly by semantic parsers or trained statistical models [121].

- *Dialog management* – dialog management module is the heart of the whole system because it designates the direction of conversation with the system. It glues all the components together and communicates with external applications, local services

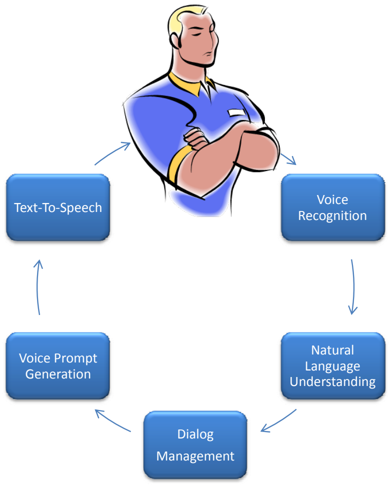


Figure 2.2: Typical internal architecture of voice-based dialog application (information flow diagram)

or remote web-based services. The dialog management module can be relatively simple rule-based system or a complex trained statistically based dialog arbiter.

- *Natural language generation* – the dialog systems need to respond in typical natural language manner. This system generates answers to user according to the decision of the dialog management module.

- *Text-to-speech* – the system's output is generated by a text-to-speech module which either converts textual utterances to speech-like sounding audio or uses prerecorded prompts.

Speech research develops techniques to process written and spoken human language. Understanding and generating natural language in a computer system requires the knowledge of several areas and it provides different views on spoken dialog systems:

- *Phonetics and phonology* – deals with how to model pronunciation of words. *Phonemes* are the basic unit of sound that can change the meaning of a word. For example, phoneme /k/ appears in English words cattle, cat, come.

- *Morphology* – analyzes, describes and identifies morphemes. *Morphemes* are the smallest grammatical units. For example, negative morpheme of "can" is "can not" or "can't". Plural morpheme of "dog" is "dogs".

- *Syntax* – studies the knowledge on how to group and order words together into sentences. For example, identifying questions, imperative sentences or declarative sentences. Structure of sentence is often visualized by parsing trees. Two main syntactic theories exist that allow to build sentence parsing trees. *Phrase structure grammars (or constituency)* was originally introduced by Noam Chomsky [27].

They are based on constituency relation. The basic structure is binary divison of a clause into subject (noun phrase) and predicate (verb phrase). *Dependency grammars (or dependency)* represent modern syntactic theory, originally introduced by Tesnière [177]. The verb is viewed as the center of clause structure. Other words are directly or indirectly dependent on the verb. Dependency trees are then flatter than constituency trees.

- *Semantics* – handles and identifies the different meanings of language. It studies relations between words, phrases, signs and symbols.
- *Pragmatics* – studies how conveying of meaning depends not only on structural knowledge. However, it depends on context of the utterance, pre-existing knowledge of both the speaker and listener. For example, it is different saying solely "No." and "I am sorry. No." The latter is more polite.
- *Discourse* – analyzes linguistic units larger than a utterance.

Further paragraphs summarize the most useful techniques for Natural language generation, text-to-speech and dialog management as they are the ones most relevant to this dissertation thesis.

2.1.1 Natural language generation

This is the oldest field of natural language processing. It investigates the way how a computer program can prepare a prompt for a user in natural language. It also explores methods on how to convey specific information to the user in the most effective way from non-linguistic input (e.g. converting program objects to user readable text).

Computers were able to generate grammatically correct sentences from the very beginning. Early, the sentences were prepared by a designer of the system and as such were not generated by the system itself. This approach of language generation is called *canned text.* There are four distinct approaches in natural language generation [178].

- *Canned text* – the simplest approach. Trivial to use in software systems. Static sentences are stored as a part of a program. The disadvantage is inflexibility, changing these static parts means changing the program. For example, "Download completed" string.
- *Template-based systems* – These systems use predefined templates and support small alterations by filling the blank spaces in these templates. E.g. a template can be the string "Good morning /name/", the /name/ variable is then filled by the correct user name (e.g. "Good morning John"). Even famous Weizenbaum ELIZA [191] system used templates. This approach is used in most of today's dialog systems.
- *Phrase-based systems* – These are also called generalized template systems. Phrasal patterns are grammatically correct structures of a target language (e.g. the object is the first and verb follows in English). Representation can be seen as a semantic tree, similar as used for language understanding in the Hidden Vector State model [71]. Phrasal pattern is selected according to the input and it is recursively expanded to

match the input. The patterns represent grammatical structure of text in a specific language. In the second step this pattern is filled by content from a database. Commonly used in language translation systems [26].

- *Feature-based systems* – this approach is based on utilising heterogeneous features of a possible generated sentences. The features can be positive/negative, may distinguish question/general sentences or present/past tense, etc. [154].

The grammatically correct generation of sentences was enough for early dialog systems that were text-based like ELIZA [191]. Spoken dialog systems need to generate speech signal from text input. This is done by text-to-speech systems shortly described by next paragraphs.

2.1.2 Text-to-speech systems

The words of generated sentences are pronounced as individual speech units that are called phones (instance of phonemes in actual utterances). The text-to-speech system needs to have a pronunciation for every word it can say. The correct pronunciations are studied by research field called *phonetics*. A phone can be defined as speech sound, phones are represented by symbols of a phonetic alphabet. The *International Phonetic Alphabet* is the standard that was introduced by International Phonetic Association in late 1800s.

There are two main approaches to generate human speech. The older one is based on the simulation of vocal tract and the second one is a data-driven approach.

- *Formant synthesis* – the method is based on simulation of human vocal tract. The sound is completely generated in software. The synthesis system changes various parameters during the synthesis process (fundamental frequency, noise level, etc.) according to a physical model and outputs sound. The system has one disadvantage that the speech sounds "robotic". This is caused by the inaccuracies introduced by the simulation of the human vocal tract.

- *Concatenative synthesis* – this approach uses speech recordings done by one human voice talent. It is based on concatenation of small speech segments. This synthesis sounds more natural than the formant in general. However, it sometimes produces audible glitches caused by variations in natural speech and by errors in databases.

The dialog management module is the central component within the spoken dialog system. This module uses information from other subordinate modules to select the correct response. This information is often called evidence. According to evidences, the expected user goal/intent and dialog history it tries to understand the dialog state and executes the next dialog turn. Subordinate modules are not 100 percent correct so the dialog manager should handle errors occurring due to incorrect/missing recognition or noises. Last but not least it should provide various kinds of grounding/confirmation strategies to confirm important assumptions with the user. The next section analyzes the dialog management module in more detail.

2.2 Dialog Management

The dialog manager finds the user intent which is then represented as semantic information, finds the answer to the user intent and outputs system response at a conceptual level. To complete a dialog cycle it uses various internal and external systems and services. One internal service can be dialog history which maintains the context of dialog, past user interaction with the dialog system and appropriate system actions. External services can be miscellaneous knowledge bases, for example web-based. These knowledge bases should provide answers to the user intents/questions, e.g. weather service which provides weather forecast for the next five days. The main role of dialog management modules comprises of the following actions:

- Find answers to the user queries by reusing internal/external databases based on the current user's utterances and the dialog history.
- If the dialog manager finds that another slot is needed to provide an answer, it would ask the user for this information. For example user wants to travel from Prague to New York, the dialog manager finds out that date of the travel is needed to complete the query, so it instructs the natural language generation module to ask for this slot.
- Confirming uncertain cases or to handle error cases.
- Predict next system action to output system's answer.
- Control conversational mechanisms (e.g. back-channels, turn-taking, multi-party dialogs).

The implementation of dialog manager can be done in various ways. The next paragraphs describe the variants of dialog management.

2.2.1 Classification of Dialog Management

Dialog can be viewed from various points of view. One view is based on participants of dialog. Special type of dialog is monologue. Monologues are defined by a speaker and a listener. The monologue is special in the way that the direction of information flow is only one, from a speaker to a listener [91]. Except control mechanisms like back-channels from the listener to the speaker.

Further in the text a *dialog* represents the conversation between humans or between a human and a computer (human-system). In this case each participant periodically takes a turn being a speaker one time and being a listener the next time.

Human-computer dialogs can be divided into three groups of dialog stated further in the text. The key important design part of spoken dialog systems is the type of dialog control strategy [116]. This choice of dialog strategy will have big impact on the system behavior, dialog naturalness and length. There are two types of spoken dialog classification. One classification divides the dialog systems according to the computer algorithm that controls the dialog [116] and the second one classifies the system through appearance of the dialog [117]. These classifications are here:

1. Finite state-based systems (Directed dialog, system initiative)
2. Frame-based systems (Mixed initiative)
3. Agent-based systems (Beyond mixed initiative)

Finite state-based systems. The dialog in this type of a system is built as a graph/state machine. The user is guided through predefined sequence of steps (dialog states). Only the system (not the user) controls the dialog flow. The system asks very specific questions and the user is expected to give very specific answers. The system begins, directs and closes the dialog. However, the rigidity of finite state-based systems is seen only in extreme cases. The vast majority of these systems allow user to switch context, e.g. using global commands like "cancel". For an example see the directed dialog with a train ticket reservation system in Example 1 [47].

```
System: Tell me the destination of your travel?
User  : Berlin.
System: What day do you want to travel?
User  : Thursday
System: What time do you want to go?
User  : At 9 o'clock?
System: How many persons will travel?
User:   ...
```

Example 1: Directed dialog example

Alternatively the system should also verify the user's input each step. The same example with "answer verification" – grounding, see Example 2 [37].

```
System: Tell me the destination of your travel?
User  : Berlin.
System: Do you mean Berlin?
User  : Yes.
System: What day do you want to travel?
User  : Thursday.
System: Was it Tuesday?
User  : No.
System: What day do you want to travel?
User  : ...
```

Example 2: Directed dialog example with answer verification

Disadvantage of the system is that the dialog is totally guided by the computer and the user is not allowed to give more information at once (in one dialog turn), s/he can only answer when the system asks. Example 1 and 2 represent typical form-filling dialogs. The system questions and user fills information in blank places. Another, even simpler dialog type is a menu hierarchy dialog. The system tells the user possible answers the question and the user only chooses the right one (example: Which system do you want to use? Flight information, banking application or reservation system?). From developer point of view there is an advantage, that the possible answers are known in advance, so the quality of speech recognition could be improved by restriction of possible words (phrases).

Frame-based systems. Frame based systems use task templates to manage the dialog. Staying in a travel reservation domain, to complete the travel reservation the system needs for example this information: departure city, arrival city, date and the time of departure. In such a frame-based system the user is not required to go through defined sequence of questions. The user can tell multiple information items in one answer and the system will ask for the remaining ones. The frame-based dialog could look like Example 3

```
System: What travel do you plan?
User  : I would like to go to Washington from Prague on Thursday.
System: What time do you want to go?
User  : at 7 a.m.
System: I have reserved the flight number ...
```

Example 3: Frame-based system dialog

Example 3 shows the main advantage of the frame-based system. The user is free to tell as many information items as s/he wants in one answer and it is up to the system to find out what information is needed to complete the task template. The implementation of this system is much more challenging, because the possible pool of answer words (or phrases) is bigger than in directed dialog. The VoiceXML standard is one example of frame-based dialog architecture [113]. **Agent-based systems**. Agent-based dialog systems go beyond mixed-initiative (frame-based systems) and contain some glimpses of artificial intelligence (AI). The systems employ AI to support planning and reasoning processes. They contain more complex models of belief and interaction. Changing the example from ticket reservation to music jukebox, suppose the conversation in Example 4.

```
User  : I want to listen to Jealous from Barbara Streisand.
System: I don’t have Jealous from Barbara Streisand. But
        I have song Jealous from Sinead O’Connor. Do you
        want to play it?
User  : Yes, play me that one.
System: ...
```

Example 4: Agent-based system dialog

Example 4 pointed out one important thing: The user does not properly remember the author of the song Jealous, but the answer of the system is not simply “NO”. The system is able to recover from this situation and finds out the right answer (similar song) and offers it to the user. Commercially this type of spoken dialog systems is starting to be used in a form of various digital assistants (like previously mentioned Apple’s Siri or Google Now, etc.). They can be used in some particular defined topics/domains. These systems are often used in very noisy environments (e.g. streets, in car, in public spaces), they need to be prepared for that and handle various kinds of errors. The next section tries to introduce general principles used to detect and recover from errors in spoken dialog systems.

Error Handling and Detection in Dialog

The performance of automatic speech recognition (ASR) modules and natural language understanding (NLU) modules have been improved, spoken dialog systems can be developed for relatively complex domains. However, there still exist problems for spoken dialog systems implementations. Errors in ASR and in NLU modules are going to be here due to the noisy conditions in which spoken dialog systems are used. These errors can cause misunderstanding of the dialog system. Dialog systems detect and repair potential errors in understanding using confidence scores from ASR and/or NLU modules [15], [104]. Most of the systems rely on manually defined confidence level thresholds. However, confidence thresholds are not a fully reliable measure, they are dependent on the environment in which the user utterances are recorded. They are also dependent on the user. So this can bring some problems that need to be solved by the dialog manager.

Skantze classified various levels of understanding in human communication into four classes [170]:

- *Full understanding*: A listener grasped the full intention of utterance.
- *Partial understanding*: Only a part of the intention was understood.
- *Non-understanding*: The intent was not understood and the listener knew it.
- *Misunderstanding*: The intent of a speaker was not caught. However, listener continued with an interpretation that was not in line with the user's goal.

Skantze found that signaling non-understanding by the dialog systems had negative effect on the user's experience. Non-understanding without signaling had no such effect. Therefore, it seems better to use an alternative error recovery solutions like various confirmation strategies [170]. The dialog management modules can use various kinds of confirmation/rephrasing strategies to recover from errors common in the state-of-the-art dialog systems:

- *Explicit confirmation* – An explicit confirmation rephrase the user intent with saying all slot values and awaits user's confirmation by simple "yes" or "no". For example in the domain of restaurant booking application the system asks question: "Do you really want to reserve table for four people at the Morton's tomorrow at 7pm"?
- *Implicit confirmation* – Implicit confirmation strategy presumes that the system automatically repeat the status of dialog to user prepending it to the next prompt. Then the user can disagree when the system misunderstood.

Approaches to Dialog management

Dialog manager should find the next system action to be executed based on the context of user's utterance and the dialog history. Three distinct approaches exist for dialog management algorithms.

- Rule-based dialog management – Most of commercial (and even academic) dialog systems are implemented by domain-knowledge experts and are authored for one specific domain only. These systems require mostly finite-state machine with hand-crafted rules. This is very inflexible. There are several examples of such systems, for example Waxholm system [21], restaurant knowledge system [92] or rail travel information system [14]

- Data-driven dialog management – This approach needs collection of dialog data and human effort in the process of annotating dialog data. Modifying the system for another domain requires new data collection and annotation only in opposite to the rule-based systems. These systems include Hidden Information State (HIS)-based spoken dialog system [59], Incremental Dialog Manager [166] or generally systems that use Markov Decision Processes or Partially Observable Markov Processes (POMDP) [195]. The data-driven dialog management systems are able to achieve nearly the same results as hand-crafted dialog systems that require much more human effort [102]. Robustness to the NLU and ASR errors is an advantage of data-driven dialog managers

- Hybrid dialog management – POMDP-based dialog manager need relatively big dialog corporas to learn an optimal policy. To overcome this disadvantage dialog managers use "simulated users" to generate such corpora. The simulated user internally contains user models that are based on real dialog system evaluations [162],[58]. The hybrid approach combines reinforcement learning techniques used in data-driven approach with supervised learning. The supervised learning limits the space needed to be explored by reinforcement learning by adding specific rules of domain knowledge. The rules can be such that a dialog system cannot reserve the railway ticket without telling a user the final price of a journey.

Conversation and dialog has been the most fundamental area of human language use. When humans need to arrange something they use dialog to communicate with other people. For example, ordering wine in restaurant, buying new clothes, in business meetings, etc.

What is characteristic about dialog? The difference between monologue and the dialog is turn-taking. Speaker 1 says something, then Speaker 2 takes the turn, Speaker 2 yields the turn to Speaker 1 and so on. This is even more visible in the multi-party dialogs. How humans know when the proper timing, when to switch roles from speaker to hearer and vice versa. This is studied in the discipline of "conversational analysis" [155], especially in the field of turn-taking and turn-yielding.

2.2.2 Turn-taking in Spoken Dialog Systems

An overview of the current state of research in the field of turn-taking/yielding in spoken dialog systems is provided in this section.

Although the recent quality improvement in dialog technologies is tremendous, as it is seen in the previous sections, dialogs with artificial systems still fall far behind in comparison with their human counterparts in terms of both comfort and efficiency. The reaction times of artificial dialogs are still slower, although the systems are improved through incorporation of turn-taking models. The model used by Raux and Eskenazi has improved latency of the system by 24% over the fixed threshold baseline [51]. Hjalmarsson also showed significant improvement of reaction time to stimuli with high agreement [77]. Nevertheless, the problem is not related only to ongoing issues in speech recognition and understanding. For example Ward et. al identified turn-taking problems as important shortcomings [189]. The dialogs between a human being and the system are typically straightforward allowing only one speaker at a time. The most common method of recognizing a turn-yield in conversation is waiting for a silent pause longer than a specified threshold. It brings one usability problem: If the user pauses inside an utterance and this pause is longer than the threshold, the user is cut off by the system [53].

Such a simple pause-threshold approach is not used frequently by humans. People tend to use turn exchanges with almost no gap in-between. That was supported by analyzes of individual human to human dialog examples [159]. However, recent work has shown that the no-gap in-between dialog turns dialog is not so common. Heldner and Edlund explored pauses, gaps and overlaps in three relatively large dialog corpora [72]. Their findings indicate that the timing of turn-taking is not as precise as often claimed. The no-gap-no-overlap model represented less than 1 percent of their data. The gaps preceding the speaker change are long enough to prepare reactive dialog control models, if given published minimal response times for spoken utterances [168].

A smooth exchange of turns between dialog partners is supported by various turn-management cues used by the speaker [43]. The turn-taking/yielding cues are sent or received before the moments when the speaker changes. These dialog spots are called Transition Relevance Places (TRPs). Further, turn-taking/yielding model is assumed as a model which is based on the general turn-taking system defined by [159]. It has only one partner talking at a time. Overlaps (more than one speaker at a time) are less than common and silence gaps are common but brief [135]. The silent partner in a dialog uses the speech cues of the speaking one to time properly his/her start of speech. These are called turn-yielding cues. For example, one vocal turn-yielding cue is a drop in loudness of speech. There is one example of a dialog sequence in the model [137] (see Example 5). P1, P2 are dialog participants and low(2-1) is continuous drop in loudness of speech. Research and identification of dialog cues started in 1970s [43].

```
P1: talks low2 low1 TRP DOES NOT TALK
P2:  DOES NOT TALK      talks talks...
```

Example 5: Example of a dialog sequence in the turn-taking/yielding model

Table 2.1 summarizes some of the turn-yielding cues with their strengths and weaknesses. The table offers an overview of many vocal turn-yielding cues thoroughly examined by scientific works. The next sections bring more detailed view on the problem of turn-taking/yielding and properties of communications cues.

Dialog Turn-Management Cues

Duncan analyzed face to face conversations in English [43]. Together with Fiske, he also stated that speakers display complex signals at turn endings [44]. These signals are divided into groups: turn-maintaining, turn-yielding and backchannels. The signals are composed of the discrete human behavioral cues explained further in the text.

Table 2.1: Table of strengths and weaknesses of different turn-yielding cues

Category	Turn-yielding cue	Strengths	Weaknesses
Vocal	Pitch fall	[43],[57] Found in 47% cases before smooth switch [67]; Correctly judged in 60% cases [77]	wide variability [35]; Found in 40% cases before smooth switch [193]
	Pitch rise	[43]; Found in 67% cases before smooth switch [193]	Found in 22% cases before smooth switch [67]
	Higher speaking rate at the end of utterance (reduced lengthening)	Significantly better z-score before smooth switch [67]	Stressed syllable means slower rate [43]
	A drop in loudness at the end of utterance	Significantly better z-score before smooth switch [67]	
	A higher frequency of jitter, shimmer	[132],[67]	
	Sociocentric sequences	[43]; Judged correctly in 90% cases [77]	Okay, yeah sequences overloaded; other cues are needed to distinguish function [67]
	Syntactical completeness (textual completeness)	[43],[161],[57],[193]; necessary cue [67]; judged correctly in 96% cases [77]	
Visual	Stopped movement of head	It's analogy to termination of any hand gestures [43]	
	Head nod at the end of utterance	Head postural shift – semantic and syntactic boundaries [94]; Head nods as backchannel request [112]	
	Eye-gaze based turn-yielding cue	Multi-party conversations [68]	Not definite role in two-party conversations [87]
Body motoric	The termination of any hand gestures	[43]	

Turn-maintaining Cues

Turn-maintaining cues are also called turn-holding or turn-keeping [48]. They indicate that the speaker intends to hold the turn. These can be vocal cues such as an intermediate pitch level, changes in speech volume, drawl on final syllable (final lengthening) and also using lengthened filled pauses (in English, e.g. “Eh...”, “Ah...”, etc.).

Turn-Yielding Cues

Speakers use turn-yielding cues in dialog to inform the listener that they have finished their turn and that the partner (partners) can take the turn and continue. These cues do not force the listener to take the floor. The listener can backchannel to push the speaker to take another dialog turn. However this scheme allows the listener to take his/her turn in response to a turn-yielding cue. Duncan discovered five vocal discrete behavioral turn-yielding cues and one visual cue [43].

- *Phrase-final intonation* – Any phrase-final intonation other than a sustained, intermediate pitch level.
- A drawl on the final syllable of a terminal clause.
- The termination of any hand gestures (visual cue).
- *Sociocentric sequences* – A stereotyped expression, like in English for example: “but uh”, “you know”.
- Drop in pitch and/or loudness of speech in conjunction of sociocentric sequence (explained in the previous item).
- *Syntactical completeness* – The completion of syntactical clause.

Backchannels

Backchannels are not cues like in previous sections but they are an important aspect of conversation management. Their function is to signal attention and interest, without interrupting the speaker [148]. The term backchannel was first mentioned by Yngve [204]. The backchannel is there defined as a message/cue sent by the listener to the speaker without the intent to take the turn. This cue is sent in order to show that listener still listens. There are several ways how to realize backchannel to the speaker [188]:

- Short expressions as “uh-huh”, “mmmh” – expressions of paying attention.
- The listener completes the sentence of the speaker (in a dialog pause).
- Requesting the speaker for clarification, explanation.
- Bodily manifestations: head nods, postural shifts, etc.

Many researchers explored the nonverbal context of backchannels in order to create computational backchannel models for generating backchannels in dialog applications [127], [74] and [40].

Turn-Requesting Cues

Wieman and Knapp identified another group of cues called *turn-requesting cues* [194]. These are used by listeners to indicate the speaker's willingness to take the floor. If they work properly the speaker ends the utterance as soon as possible. The requests are very frequently accomplished by simultaneous talking. It has been experimentally shown that these turn requests are likely to be displayed if the speaker has the turn for more than twenty seconds [119].

Properties of Turn-Management Cues and Models of Turn-Taking

The turn-management cues could possibly be culturally sensitive. Ward and Bayyari described the experiment where English speakers tend to misinterpret Arabic cues [188]. However, recently Carlson and Hirschberg found that predicting phrase boundaries was fairly well recognized by non-native speakers; Japanese and Chinese native speakers who do not speak Swedish were able to easily distinguish turn-ends in Swedish dialog [20].

If all these turn-management cues could be incorporated into speech dialog systems, the dialog manager would make faster turn change decisions and this would possibly lead to smoother dialog between the human and the system. Fortunately, not only would the input part of the dialog system benefit from the turn-taking mechanism. The turn-management could also be applied in the speech output. The output module might produce turn-yielding signals when the system is about to finish the utterance and the user is waiting to take a turn.

Several generative computational models of turn-taking were proposed and analyzed. This includes complex work by Thrisson on construction of the Ymir turn-taking model (YTTM) [180]. This model represents completely automated perception of a dialog participant behavior, including speech, prosody and body language and generation of animation and speech in real-time without using manual force. The YTTM model was further extended with a system that can learn turn-taking skills from a partner using machine learning [90]. The learning system was experimentally evaluated using human participants in spoken interviews conducted by the dialog system [89].

There are works that study impressions from an agent that uses different turn management strategies. Maat and Heylen demonstrated that it is possible to create different impressions of an agent's personality by modifying the timing of turn beginning and by utilizing overlapping speech using their conversational simulator [173]. The work was extended and supported by perceptual experiments with recorded conversations between human and interviewing agents [174].

Recent works further exploit findings of turn-taking and analyze the relationship between turn-taking and yielding cues and turn boundaries during perception experiments. Oliveira and Freitas present turn-taking prosodic issues in a telephony environment by analysis of judgments of non-participating dialog listeners [133]. Oliveira and Freitas found out that analysis of dialog out of the context is problematic and should be avoided [133]. Gravano describes an extended number of prosodic phenomena occurring in TRPs [67]. The work by Hjalmarsson introduces a perceptual experiment which compares human dialog turn-taking signals to synthesized turn-taking signals and confirms the correlation between the number of signals and faster turn-taking decision times [76].

All previous works and experiments using some sort of human-like agent consider mainly prosodic (vocal) turn-management cues. There are several papers that research multi-party dialogs and participants' eye-gaze as very important turn-taking/yielding cues. Notable works include development of a visual attention computational model [68] or multi-party eye-gaze experiments [88], [87]. Padilha and Carletta addressed also small group discussion turn-taking behavior [137]. They compare base (only vocal cues) and extended (vocal & visual cues) model of turn-taking using a multi-agent system. Multi-party turn-endings prediction was also examined [39]. Their model learned from real multi-party meetings but the results were not so good, because the best F_1 score of this model is lot lower than in the case of [163].

The research of turn-taking that was addressed in this section shows that there are some promising ways in introducing turn-taking/yielding into humanized user interfaces, not only in speech channel but also incorporating some turn changing cues into human-like agents. The next section surveys the relevant literature of human-like conversational agents.

2.3 Embodied Conversational Agent Interface

Voice-based interfaces is one part of interaction techniques improvement. Second part uses for example the face-to-face communication paradigm. Seeing virtual faces also humanizes the computer-user interface and makes it well acceptable for common users. By expressing emotions, faces enrich the user experience even further [12], [17].

Embodied Conversational Agent (ECA) is the user interface metaphor that allows to naturally communicate information during human-computer interaction in synergic modality dimensions, including voice, gesture, emotion, text, etc. Due to its anthropological representation and the ability to express human-like behavior, ECAs are becoming popular interface front-ends for dialog and conversational applications.

Figure 1.8 shows an example of such an ECA interface that anriches a classic GUI. This agent is a central part of a developed voice-based dialog application that is used to handle playing of MP3 music files.

Adding ECAs modifies Figure 2.2 and adds a new component that takes care of generating the ECA behavior and rendering the agent UI. The user speaks to the microphone and speech recognition module transforms speech to sentences in text form. These sentences are then processed by a natural language understanding module, which tries to find the user intent. The user's intent is processed by dialog management module. This module uses others systems to prepare the answer for a user and then synchronizes talking agent behavior (speech, gestures) and means of the GUI to produce a coherent answer (see Figure 2.3).

The main task of ECA research is to create an intelligent agent that is capable of social behaviors that are typical for humans. The work on these agents is inherently interdisciplinary. Implementing ECAs requires research in disciplines from psychology of humans [50], [101], over natural language processing, speech synthesis, dialog management (see Section 2.2), to agent architectures [164], [181], over emotion architectures, to interface design.

ECA research attempts to solve the whole problem of building a conversational agent. This approach has a weakness in that building the complete agent takes most work and behavior specialties are hard to address [23]. This leads researches and developers to

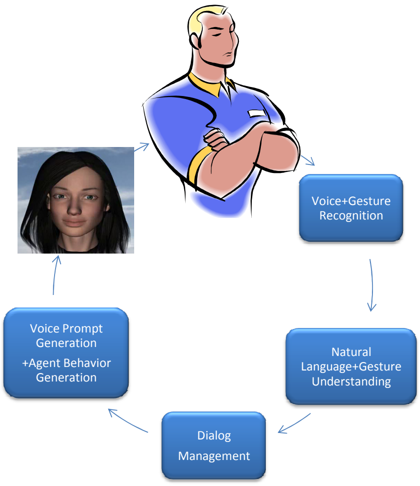

Voice+Gesture Recognition
Natural Language+Gesture Understanding
Dialog Management
Voice Prompt Generation +Agent Behavior Generation

Figure 2.3: Typical internal architecture of voice-based dialog application with humanized user interface (information flow diagram)

prepare toolkits and ECA behavior authoring languages that address the problem of using a talking agent in applications.

2.3.1 Talking Agent Frameworks

The authoring environments for talking agent-based applications are fragmented, and researchers tend to develop their own languages. The existing languages tend to fall to different categories of abstractions, accenting the respective design priorities that the authors were following in support of their projects. One class of languages supports the modeling of human behavior at a very high level, such as Human Markup Language [131]. HML is a standardization effort that tries to use XML to express human behavior, e.g. emoticons. It provides means, that tags various verbal and non-verbal communication cues used in human-to-human interactions. Its complex design makes it difficult for authors to get down to the level of plain animation when needed.

What seems to be the prevailing concept in ECA behavior authoring language design is the notion of independent communication channels. These channels, such as head, speech, gestures, body, expressions, etc. are mixed and matched to a multimodal communication act that the talking agent as the "anthropological" output device delivers to the user. Examples of languages supporting the channel mixing concept include VHML [109], SMIL-AGENT [10] and RRL [145]. The RRL language was further developed in the NECA project [185].

The Web application domain has brought its own set of XML-based languages that help Web page designers enhance human-machine interaction experience. Multimodal Presentation Mark-up Language (MPML) [150] builds on the body of the Microsoft

Agent to create predefined animation sequences. Behavior Expression Animation Toolkit (BEAT) [22] processes the XML input description in the form of a tree containing both verbal and non-verbal signals, to produce the synchronized animated sequence on the output.

Relatively large part of work in the area of ECA authoring languages was done in the SAIBA (Situation, Agent, Intention, Behavior, Animation) framework, e.g. Functional Markup Language (FML) [75] or combined communication language FML-APML (Affective Presentation Markup Language) [108]. Recently, Expressive Multimodal Conversation Acts (EMCA) communication language was developed [157]. It is based on so called Expressive Speech Acts, which allows for describing complex emotions of conversational agents.

Hong et. al. proposed an XML-based interactive drama markup language (IDML) [78]. This language is interesting in the way it supports concept of branching ECA narrations. This allows users to interact with an ECA and enable them to affect ECA's behavior. Authors created a visual tool to build IDML scripts to speedup the language learning path.

Some authoring environments enable developers to animate characters in realtime and they allow them to fine-tune behavior on the layer of specifying partial key poses in absolute time, e.g. EMBRScript [73]. This can be useful for animators, but not for spoken dialog developers.

Most of these languages and toolkits tend to split the application authoring task into (a) off-line step for preprocessing and (b) the real-time step of running the preprocessed animations. This comes with the implicit disadvantage, that the animations are realized on the closed set of precomputed behaviors.

The agents take an appearance of human body or face only displayed on the computer screen or using a projector. The detailed review of current state of ECA frameworks and technologies was provided in [4].

The simplest kinds of ECA are agent systems that are not interactive. The ECA is here used as a simple presenter to accompany the speech targeted for listeners. This includes the framework *PPP persona*, which was created for guidance in web-based applications. It uses pointing gestures and talks through a speech synthesizer [5]. This work was further extended to the AutoBriefer agent, that can automatically generate presentations anchored by an animated agent [46].

This paradigm of using ECA as presenter is sometimes extended by using a group of agents that convey the message. This method is borrowed from TV news, where for most of the time news are anchored by two presenters. This approach preserves the non-interactivity with human users. Example of a system with multiple ECA agents is the eShowroom demonstrator [158]. The application developer can script dialog sequences between ECAs to present properties of a new car in the car showroom.

The most common version of ECA frameworks are the frameworks suited for one ECA interacting with one human. The REA (Real estate agent) was developed with conversational naturalness in mind. The ECA in this system represents a real-estate agent that tries to sell houses to users. The agent is full-bodied and uses gestures and expresses emotions [23]. ECAs were also tried in medical environments, the Greta agent is the most famous [125]. This ECA agent uses synchronized speech and gestures. Well known are also works by Thorisson Gandalf ECA systems; built-in turn-taking models are very interesting in these systems [180].

Some ECA frameworks even try to tackle multi-party conversations. For example project IDEAS4Games developed a Poker game [60].

The important component of ECA agent frameworks is the speech visualization component and appearance of life-like ECA agents. In the next paragraphs, evaluation methods of speech visualization and dynamic and static appearance of talking agents will be addressed.

2.3.2 Speech Visualization Evaluation

The primary goal of ECA articulation research is to produce realistic visual articulation which is indistinguishable from that of real humans. This task includes implementation methods and in the end evaluation of results.

The quality of talking heads visual articulation has been measured using various methods. These comprise subjective evaluation, perception of speech in noisy environments and others.

The most common method is subjective evaluation [32]. This method is based on getting comments and rating from naïve and expert participants. This provides information on quality of talking head articulation but does not give possibility to compare various talking head implementations.

The second method - perception of speech in noisy environments improves the quality of results and facilitates comparisons of miscellaneous talking head implementations. Participants try to listen to talking head footage in a noisy room and the ability to understand the words indicates how much they are able to improve the intelligibility of speech by lip-reading the talking head [111].

Good results are provided by methods based on a forced choice. Participants see videos and identify which animation is real or synthetic [69].

Extensive experiments of the McGurk effect (introduced in Section 1.1.3) using talking head were done in [110]. An interesting method based on perception of this effect was proposed in [33]. Participants are given synthetic talking head McGurk sequences and their confusion responses are measured and evaluated. The report does not mention whether participants had corrected vision or not.

Apart from speech visualization, other characteristics of ECA agents are need to be addressed to create life-like agent presence. The next section focuses on static and dynamic behavior characteristics of talking ECA faces.

2.3.3 ECA Behavior Parameters Evaluation

Trying to implement proper full human behavior is very expensive. Part of agent's visual behavior is its appearance. The problem of appearance is often solved "statically." The user could setup his agent appearance by modifying parameters as for example color of eyes, color of hair or size of nose, etc. But there is another possibility to change appearance – reflecting "dynamic" part of appearance, for example mouth opening, hair length (dynamic hair movement), head movements, etc. Let's give the user of a dialog application possibility to choose proper dynamic behavior. As in real life, where listening to people whose behavior (appearance) we like is more pleasant, proper settings of agent's parameters could lead us to creating more enjoyable agents and thus to improved communication with an agent. Therefore it could be worth of creating a method how to find out human sensitivity to these “dynamic” parameters.

Developers of ECAs want to produce an intelligent agent that can draw user attention and is capable of certain social behaviors. Measuring the achievement of this goal uses findings and procedures introduced by usability testing [124]. Evaluation of an ECA application depends on the stage of system development. One could test scientific ideas and hypotheses; for this purpose s/he should choose a proper test methodology.

As described in [28] there are four big research lines in ECA development.

- The first one is *understanding the world*. This line sets questions like: Does the ECA interface ease communication with the application? Is the user satisfied? For example, agents could increase quality of learning more than a voice-only application [106]. But they can also impose more and different kinds of cognitive load on the user [105].

- The second possible research line is *towards a target application*. This means evaluating the use of agent for a specific purpose. An example could be a virtual guide agent in virtual environments [179]. An evaluation method of agent's particular actions is described in [172].

- The third evaluation method checks *conformance to standards*. Although the ECA field lacks proper standards, there are some that are accredited and thus should be taken into account, i.e. MPEG4.

- Last, but not least, another testing methodology prefers *comparison of designs*. This method examines human behavior when engaged in interaction with an ECA. To exclude the influences of agent's appearance, Xiao et. al. used three types of agents (animated, stiff and iconic) and compared the reactions of users [200].

To enable ECAs to express proper behavior, various knowledge bases are needed. To make these knowledge bases interoperable between systems, it is advisable to use ontologies to describe the knowledge bases. A domain ontology can describe possibilities of ECA's behavior channels, emotions, parts (face, body, arms, legs,...). Moreover, these ontologies can describe means of the whole system's environment (including UI capabilities). The domain ontology can also describe behavior patterns that depend on various environmental, user and application contexts. The knowledge base containing these behavior patterns can be used as a source when rendering an ECA utterance. The next section tries to describe the significance of ontologies for ECA development.

2.3.4 Ontologies

Ontologies describe the structure of knowledge bases. Firstly, we need to define *knowledge*.

Definition of Knowledge

Knowledge could be defined [1] as:

1. expertise, and skills acquired by a person through experience or education; the theoretical or practical understanding of a subject,
2. what is known in a particular field or in total; facts and information,
3. awareness or familiarity gained by experience of a fact or situation.

Further, knowledge is understood as representation of what is known in a particular domain of interest. From a human point of view, acquiring knowledge involves a complex process of perception, learning and reasoning.

From a computer point of view, the idea of knowledge is related to *the knowledge base*. The knowledge base is a computer representation of particular domain knowledge. The knowledge base can be seen as set of domain ontology instances too. Same as a domain ontology describes the structure of knowledge, domain knowledge represents all the knowledge in a domain.

Definition of Ontology

Ontology could be defined either in the field of computer science or in the philosophy field.

In philosophy, the word ontology comes from Greek and is part of study of metaphysics. Ontology itself is study of being, existences and reality. This study investigates living entities and their grouping and possible hierarchies [147].

The philosophical meaning of ontology has some commons with the computer science way of ontology definition. They both study entities, relations between them and their possible categorizations.

In the context of computer and information sciences, an ontology defines a set of representational primitives with which domain knowledge is modeled. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application (To appear in [136]). Figure 2.4 shows definition of ontology from Artificial Intelligence (AI) point of view, with complexity level and relation to automatic reasoning systems [171].

Computer applications deal with various and big amounts of data. These data structures sometimes contain structured knowledge. Working with from complex knowledge comming from various sources would be rather difficult without ontologies. They provide common terminology when addresing differently managed knowledge bases. Firstly, an ontology describes knowledge structure for a particular domain of interest (e.g. domain of music, medical diseases, computer hardware, human genes). A domain of interest is the space which the ontology describes. Secondly, an ontology allows to share knowledge between heterogeneous systems that adopt the same ontology. Heterogeneous systems are distributed systems that contain different kinds of hardware and software in cooperative fashion to solve problems [18].

Use of Ontologies

Noy gives several reasons why to develop ontologies [129]:

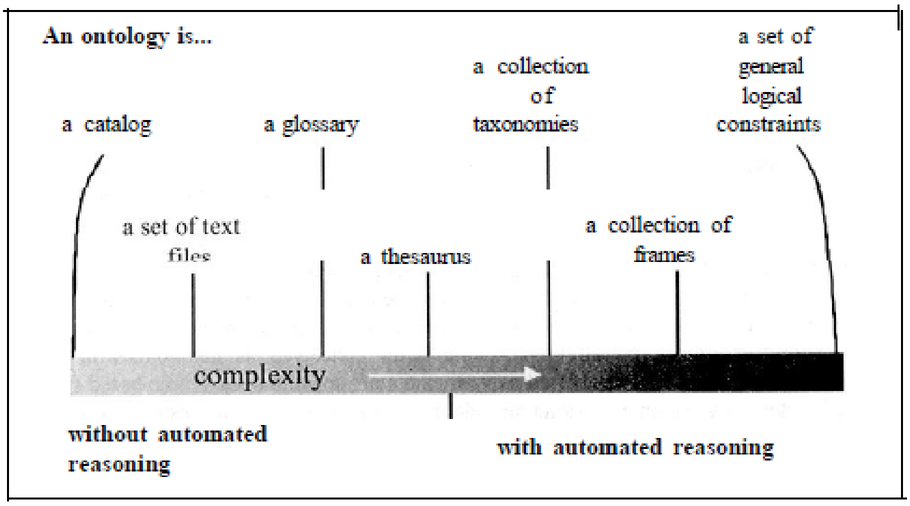


Figure 2.4: Definition of ontology from AI point of view. (Source [171])

1. To share common understanding of structure of information among people or software
2. To enable reuse of domain knowledge
3. To make domain assumptions explicit
4. To separate domain knowledge from operational knowledge
5. To analyze domain knowledge

Sharing common understanding of structure of information is one of the most used reasons to develop an ontology. For example, a software program working in a specific domain (e.g. searching offers of flight tickets and finding flights according to constraints for users). Suppose such program uses a database of flights from a single airline. Imagine upgrading this program to offer flights from a new airline. However, the database of the new airline is different from the currently used database. Without a defined domain ontology, developers need to adjust the program code to be able to understand the structure of the new database. Having domain ontology defined, the new airline database can be added without hassle with program code because the data structures are defined by a shared ontology.

Reuse of domain knowledge is another advantage of usage of ontologies. Sometimes ontologies need to understand some common things as for example the calendar, in order to enable the software know how the days (months, years, ...) go. It is just enough to build this calendar ontology once and another future designed ontologies will reuse it.

Making domain assumptions explicit helps developers of software. They should not hard-code domain assumptions to application code, instead they should use an ontology. If the assumptions change, it is very quick even for an advanced user (not the system developer) to change these assumptions without changing the source code.

Separating the domain knowledge from operational knowledge is another common use of ontologies. A task of configuring a software product from its components can be described according to a required specification and a program can be implemented that does this configuration independent of the product and components themselves [114].

Analyzing domain knowledge is valuable when a developer wants to reuse or extend ontologies. Before starting analysis of domain knowledge, basic concepts of the analyzed domain should be defined.

The list above mentions rather advatages of ontologies and why to use them. However, ontologies are very complex structures. Thus, it takes significant amount of time to build a good domain ontology. Even a relatively small domain ontology can be really complex and difficult to maintain. Hence, ontologies are not widespread in software products. This slowly changes and ontologies are developed in collaborative environment, e.g. Freebase ontology [16].

Ontology structure

The structure of ontology is given by how an ontology is represented (the most common is a formal ontology language). But there are some similarities between these representations. They result from the ontology definition in section 2.3.4; an ontology describes concepts in a particular domain. Commonly, ontology consists of the following components [129]:

1. *Concepts* (or classes, sets, collections) – Basic building blocks of ontology.
2. *Attributes* (or slots, roles, properties) – Properties and features of each concept.
3. *Facets* (or role restrictions) – Restrictions defined on attributes.
4. *Instances* (or objects) – Instances of concepts. Set of instances creates a "knowledge base".
5. *Relations* – Define relations of concepts or instances.
6. *Subclasses* – A specific type of relation that expresses specialization/generalization between two concepts.
7. *Axioms* – Assertions applied to the content of the whole ontology. They are defined to ensure consistency of the ontology.

Concepts define the structure of an ontology. For example, imagine defining components for the domain of ECA. The main concept will be an agent (ECA). This concept will have its subclasses e.g. software agents, hardware agents (robot), etc. Subclass software agent can be further divided into human-like characters or fantasy characters. See the whole scheme in the fig. 2.5.

Ontology Example

Wine Ontology Example. In this section, a short example of building a partial ontology of wines is shown. Concepts are the focus of most ontologies. Concepts describe relevant objects in the domain. Concepts will be differentiated by italics in the following text. For example, a concept of *wines* represents all wines. Wines are instances of this concept, e.g. the Bordeaux wine is an instance of the concept of Bordeaux wines. A

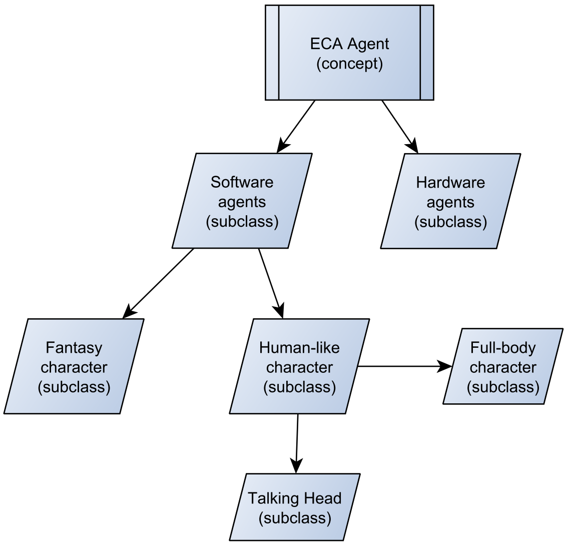


Figure 2.5: Partial hierarchy example of ontology concepts – ECA domain

concept can have subconcepts that represent concepts that are more specific than the superconcept. For example, dividing the concept of all wines into *red*, *white*, and *rose wines*. Alternatively, a concept of all wines can be divided into sparkling and non-sparkling wines [129].

Attributes describe the properties of concepts and instances: Chateau Lafite Rothschild Pauillac is produced by the Chateau Lafite Rothschild winery (see Fig. 2.6). There are two attributes describing the wine in this example: the attribute body with the value full and the slot maker with the value Chateau Lafite Rothschild winery. At the concept level, it can be said that instances of the concept *Wine* will have slots describing their flavor, body, sugar level, the wine maker, etc.

All instances of the concept *Wine*, and its subconcept *Pauillac*, have a slot "maker" (producer) the value of which is an instance of the concept *Winery* (Figure 2.6). All instances of the concept *Winery* have a slot produces that refers to all the wines (instances of the concept *Wine* and its subconcepts) that the winery produces.

The whole development process of an ontology (e.g. wine ontology) consists of these steps:

- Define concepts in the ontology.
- Develop taxonomic concepts hierarchy.
- Define attributes and describe allowed values for these slots.
- Fill in particular values for attributes for instances.

The ontology building principle can be used to build domain ontology of ECA behavior patterns that can be employed as high-level building blocks when preparing an ECA dialog turn.

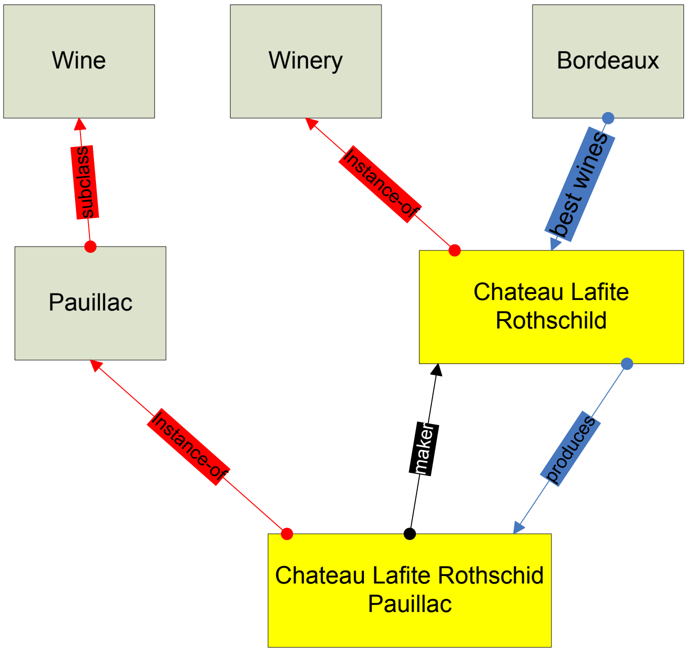


Figure 2.6: Example of a wine ontology. Some concepts, instances, and relations among them in the wine domain. Yellow color is used for instances and gray for concepts. Line defines relationships between objects.

2.4 Summary

The speech-enabled applications were introduced in this chapter to show which modules are needed to build a dialog system. A typical speech dialog application is composed of the following modules (see Figure 2.2):

- *User input*
- *Automatic speech recognition*
- *Natural language understanding*
- *Dialog management*
- *Natural language generation*
- *Text-to-speech*

Methods for natural-language generation and text-to-speech are briefly described. Dialog management approaches are thoroughly reviewed and finite state-based systems dialog examples, frame-based systems and agent-based systems are also shown here. Frame-based and agent-based systems seems to be among the most promising ones according to options for users. Finally, dialog management algorithms are covered by the first section in this chapter, including statistical data-driven systems.

Further, this section reviewed the state of the art in the research of turn-taking. Turn-taking, turn-yielding and backchannel cues are defined here. Multiple turn-taking models for speech-based dialog applications are reviewed. Turn-taking/yielding cues use the speech channel nowadays. Reviewing the literature, it seems that visual turn-taking/yielding cues can be another possibility to express turn-taking or turn-yielding in conversational agents.

In the second part of this chapter, embodied conversational agents (ECAs) have been introduced. The advantages of ECAs are emphasized, as they are means to express human-like behavior (e.g. emotions, naturalness). ECAs provide a valuable extension of speech-based dialog systems. Several ECA applications/frameworks have been listed in this section. At the end of the section various approaches to the evaluation of humanized interfaces are given, especially for speech visualization and for "static" and "dynamic" appearance parameters.

Last part of this chapter defines and describes ontologies. At the end of this section, ontology development process is illustrated using the wine ontology. Ontologies represent a valuable tool for domain specification. The provided means can be used to allow interoperability between heterogeneous systems and systems with different terminology.

Although the research in the fields of humanized user interfaces is very broad and thorough, problems exist in this broad field. Some of them are addressed in this dissertation thesis. These problems include:

- visual turn-yielding cues for ECAs.
- combination of "classical" GUI interfaces and talking agents
- evaluation of "static", "dynamic" appearance parameters
- evaluation of visual speech using McGurk phenomena
- ECA behavior patterns using ontologies

The next chapter gives better insight to these issues and proposes solutions. The issues in humanized user interfaces are examined from the "Billionaire" serious game point of view.

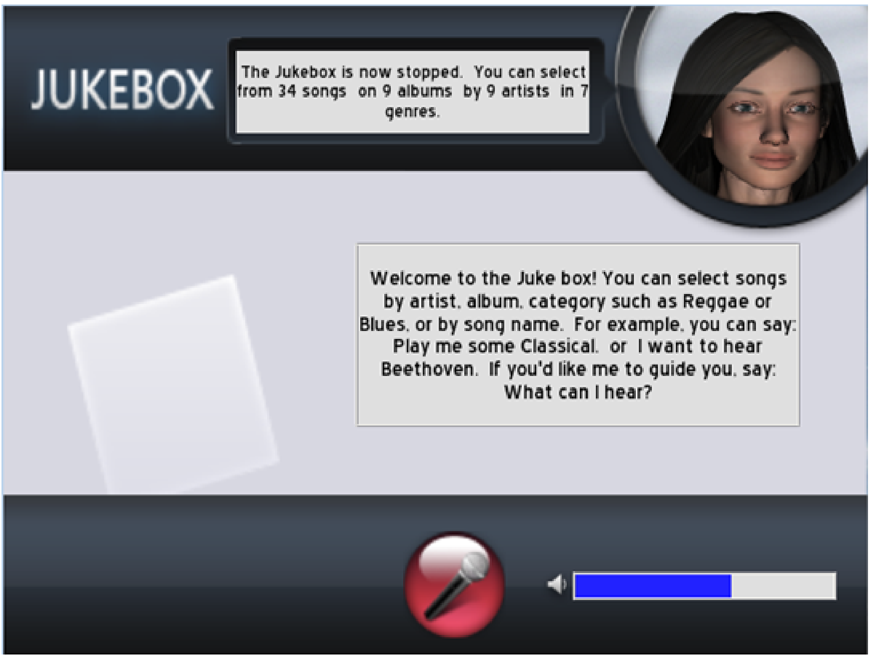
3 Thesis Contribution and Solution Proposal

3.1 Issues in ECA-based Communication with User

As mentioned in the previous chapter the research in the field of ECAs is extensive. Humanized user interfaces can in some specific environments bring more naturalness in communication with spoken dialog systems. They provide means to express human-like features, like emotions, mood and other modalities added to speech channel.

ECAs are rather useful in areas of software applications that provide education, in practicing acquired skills or as a virtual characters in games. This environment provides self-evident role for ECA as a teacher, partner or moderator. The ECA enriches the speech channel produced by spoken dialog systems and partly can solve the anxiety of users to speak to computer programs. However, there are environments where ECAs are not a suitable form of user interface. Highly productive applications (office systems, modeling software, etc.) are much more likely to be controlled by keyboard/mouse or tablet interfaces. The ECA may be seen as an element that slows down interaction with these user interfaces.

Nevertheless, ECA research tackle large set of problems, there are places where it is possible to contribute and solve problems. One issue worth of solving is the naturalness of a dialog with an ECA. Turn-taking and turn-yielding offers a possibility to solve some of these problems. The area of turn-taking and turn-yielding is investigated mainly from the spoken dialog systems point of view. The proposed turn-yielding cues are mainly vocal. One goal of this dissertation thesis is to research facial turn-yielding cues that can be used in various ECA interfaces. Using a broader set of turn-yielding cues can help creating an ECA that feels more natural in interaction with users.



The Jukebox is now stopped. You can select from 34 songs on 9 albums by 9 artists in 7 genres.

Welcome to the Juke box! You can select songs by artist, album, category such as Reggae or Blues, or by song name. For example, you can say: Play me some Classical. or I want to hear Beethoven. If you'd like me to guide you, say: What can I hear?

Figure 3.1: Example of combination of ECA and “classical” interface – ECA is depicted in the upper-right corner and classical interface building blocks like text boxes, progress bar and icons covers the rest of UI area

The second class of problems is situated in the way of combining "classical user interfaces" and ECA user interface. The authoring toolkits and control languages for ECA user interfaces are fragmented. When the designer of a system wants to build up a system that is a combination of ECA and user interface like buttons, texts, images, videos, etc. (see Figure 3.1), s/he is pushed to the situation to use one authoring language for the "classical" interface and one language for the ECA interface. However, to deliver smooth performance of this kind of user interface one should synchronize these two channels. This and the issue of generating/using proper behavior patterns for ECA are solved in this dissertation thesis. The proper behavior patterns means controlling ECA in a human-like manner based on various context variables. For example, presenting a news article by an ECA includes a set of interaction rules than differs from simply prompting the user "Do you want to continue?" Also, surrounding environment can influence the interaction with an ECA. Kiosk-based application in public space needs to handle multiple users in contrary to an ECA used as a personal assistant that is bounded to one particular user.

The last contribution area of this thesis consists of various improvements in the evaluation methodology. The whole Chapter 5 is dedicated to this problem.

The following paragraphs will describe the contribution of the dissertation thesis and solution proposals to problems defined in the Introduction part of the thesis.

3.2 "Who Wants to Be a Billionaire" Game

The solutions that this thesis proposes will be tested and used in development of a real world game "Who Wants to Be a Billionaire". This game that includes an ECA as moderator has been developed under the umbrella of the Netcarity EU project `http://www.netcarity.org`, which is described in Section 1.1.2.

Speech-enabled Billionaire game is motivated by the TV show "Who Wants to Be a Millionaire"? (see Appendix D for brief explanation) with entertaining nature of the application, even more supported by the novelty of using the speech interface. Main motivation of the game is in measuring responsiveness, mental abilities and reaction time of its user and comparing it with the behavior pattern collected during previous runs of the game.

The game has been built with an ECA in mind. The ECA works here as a natural moderator of game progress. User progression in the game is described by a state diagram (see Figure 3.2). The player during a game answers questions until a wrong answer is reached or until the game ends. Brief game scenario is composed of three steps:

1. Start of the game.
2. Answer questions until a wrong answer or until the game arrives to an end.
3. End of the game.

In each round, a user is offered four possible answers and the user should choose one. If s/he is not certain about the answer, help can be used. There are two possibilities of help: "50:50" and "audience help". The help "50:50" means that two wrong options are discarded and the user should choose from the two remaining. The help of audience is that the "simulated audience" is asked for help and the result is presented to the user.

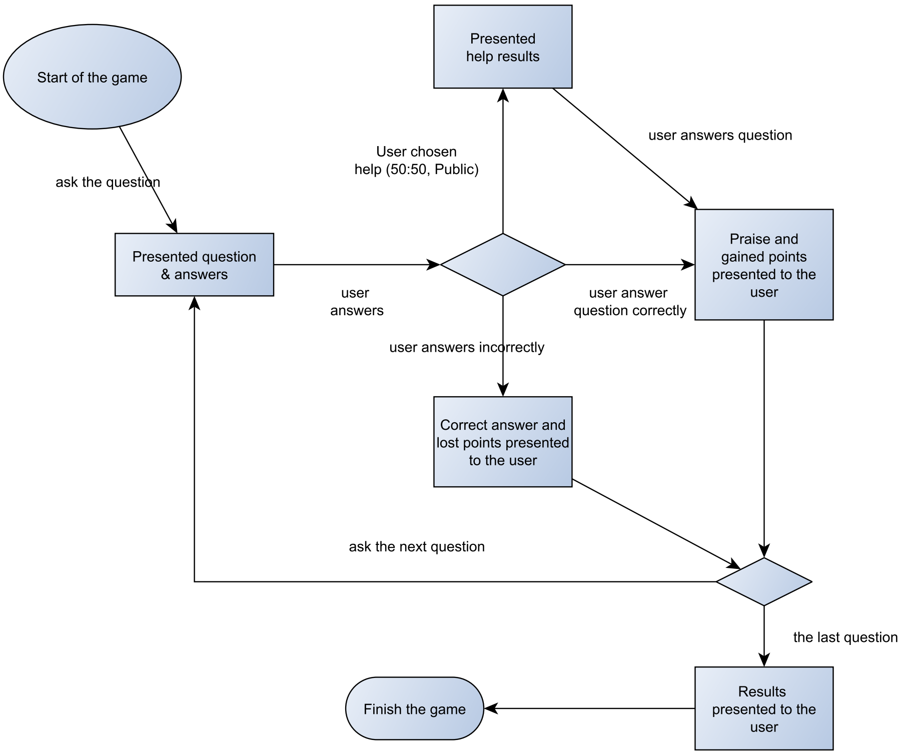

Start of the game
ask the question
Presented question & answers
user answers
User chosen help (50:50, Public)
Presented help results
user answers question
user answer question correctly
Praise and gained points presented to the user
user answers incorrectly
Correct answer and lost points presented to the user
ask the next question
the last question
Results presented to the user
Finish the game

Figure 3.2: The flowchart diagram of Billionaire game progression

The task of one round is depicted in a hierarchical task analysis diagram (Figure 3.3. The round begins by ECA asking the first question and offering four possible answers. User chooses one of the answers or asks for help. ECA then repeats the user's answer and the user confirms the answer. After the answer is evaluated, the user is told the result together with the correct answer.

The game is from the beginning designed as a truly multimodal application. It uses ECA and speech synthesis with speech recognition. However, the user is allowed to switch modalities and to use the touch interface or keyboard to answer questions.

The Billionaire game has specific demands on the graphical user interface design because of possible multimodal interaction. Figure 3.4 depicts the first prototype sketch of the Billionaire GUI. Notable parts of the GUI are:

1. Area with question
2. Area with possible answers
3. Area with game progress
4. Area with ECA moderator
5. Area with help
6. Microphone status (voice user interaction)

As it is a common practice in the state-of-the-art dialog applications, Billionaire speech recognition is started by the user pushing a push-to-activate button. This is not very

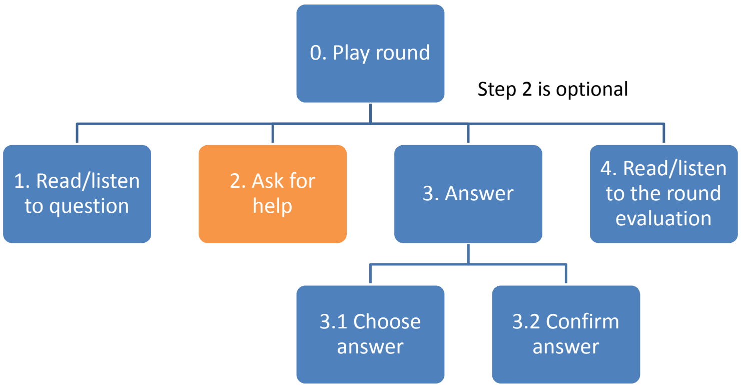
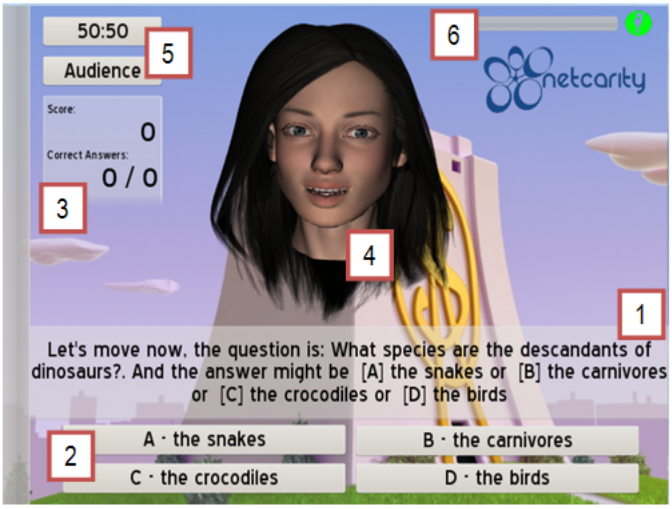


Figure 3.3: Hierarchical task analysis of one round of the game. Step 2 (Ask for help) is optional



Let's move now, the question is: What species are the descandants of dinosaurs?. And the answer might be [A] the snakes or [B] the carnivores or [C] the crocodiles or [D] the birds

Figure 3.4: The first GUI concept of Billionaire game GUI

natural and it forces the user to use the keyboard before s/he utters an answer. The push-to-activate button can be very confusing, especially for novice users. During interaction with speech-enabled programs, they tend to forget releasing the push-to-activate key or they push the key in the middle of their utterance. These errors then translate to an incorrect speech recognition result and thus to wrong behavior of applications. The following section outlines another way to yield turn from moderator of the game to the user.

3.3 Visual-based Turn-yielding

Today's spoken dialog systems predominantly use concatenative text-to-speech system to generate human-like speech. Unfortunately, concatenative speech synthesis which essentially uses chunks of human recordings, cannot often simply modify these prosodic voice parameters without distorted output sound. The formant speech synthesis does not sound like a human, but the prosodic voice parameters (pitch, speed, etc.) can be simply modified. So there is a possibility to incorporate vocal turn-yielding in formant speech dialog systems.

However, spoken dialog systems that are using ECA as user interface have many more output channels for transmitting cues to the user. Experiments that would compare visual and vocal turn-yielding cues on ECA are rather sparse. Mostly, visual turn-yielding cues are explored in multi-party conversations, especially the gaze turn-yielding cue [87]. For ECA-based systems, a model of communication like that depicted in Figure 3.5 is presumed. The model is a simple one that allows one participant talking at the same time (system or user), it can be seen as passing speech token between the system and the user. It omits the case of interrupting someone's speech which is not solved. Taking into account efficiency of communication, the simple push-to-talk technology could be very efficient in a speech dialog system as suggested in [52]. But, on the other hand, it loses its interactive ability and can be associated with a "vending machine" syndrome [180].

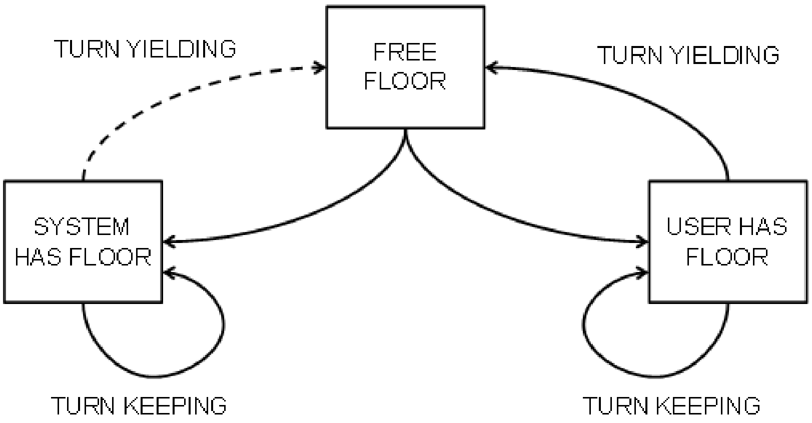


Figure 3.5: Diagram of communication between user-system. The dashed line represents main focus of system turn-yielding research presented in this thesis.

The Billionaire ECA-based game mentioned in Chapter 1 is a single-player game. Our area of interest is narrowed because spoken dialog applications involving only one human user and a computer (two-party) are different from a multi-party approach in terms of turn-taking/yielding cues (smaller space, intimate interaction, importance of some cues) [87].

The research of turn-yielding is divided into two parts. The first part focuses on the number of cues used by dialog participants and the second one tries to asses the quality and impact of selected visual turn-yielding cues in comparison to vocal turn-yielding cues. Two hypotheses based on these interesting areas are introduced further in the text.

The first hypothesis focuses on the number of cues used regardless of whether the cues are vocal or visual. That is in line with Duncan's work [43]. The higher the number of atomic turn-yielding cues at one point of conversation, the higher the probability of a correct judgment. This reasoning was also supported by experiments [77].

Hypothesis H1 – Using more turn-yielding cues before a transition relevance place increases the probability of the correct judgment about the next speaker. The turn-yielding cues can be both vocal and visual.

Table 3.1: Table of proposed turn-yielding cues

Category	**Turn-yielding cue**
Utterance pitch (vocal)	Fall
Utterance final speed (vocal)	Higher
Utterance final loudness (vocal)	Low
Talking head movement	Stopped
Talking head nod	Head nod

Second, it would be beneficial to learn about the impact of selected visual turn-yielding cues on human judgment in comparison to a selection of the vocal ones. Expectations of variation in terms of such impact are based on the fact that while one dialog partner listens to the other, his/her hearing system is occupied by hearing the partner's speech. However, the human visual system, which has even more capacity than the hearing system [19], is not occupied.

The relation between auditory and visual turn-taking cues using recorded human conversations is studied in [11]. They systematically compared audio-only, vision-only and audio-visual groups of cues. Interestingly, they found out that audio-only cues are less reliable than other groups. So, it is assumed that visual turn-yielding cues are more reliable in the process of yielding a turn in a dialog.

Raux and Eskenazi presented another interesting result. Also an implementation of an end-detection algorithm revealed that prosodic features did not help turn-ending detection once other features (semantic) were included [153]. Duncan finds the termination of any hand gestures as an important turn-final visual cue [43]. These results lead us to the following hypothesis:

Hypothesis H2 – Visual turn-yielding cues are better than vocal cues in increasing the probability of a correct judgment of who will be the next speaker.

The realization of turn-yielding mechanism in our talking agent architecture introduces several turn-yielding cues. The selection consists of three vocal turn-yielding cues (pitch fall, higher speaking rate and loudness at the end of utterance) because of their effectiveness (see Table 2.1) and because they are also re-synthesizable. Two visual turn-yielding cues introduced to the architecture are:

- movement of head is slowed down before yielding the turn in dialog
- small head nod at the end of utterance

They were selected because they can be seen on the face, i.e. the whole body is not needed for their expression. An eye-gaze based turn-yielding cue was not selected. As stated in the literature, eye-gaze based turn-yielding cues are very important in multiparty conversations [68] but they do not have as definite a role in two-party conversations [87]. Turn-yielding cues are summarized in Table 3.1 and they are more thoroughly described in the next chapter.

The proposed turn-yielding mechanism needs to be incorporated into the talking agent architecture. As part of the ECA architecture, the system can prevent some of the errors that are introduced by push-to-activate buttons in some specific cases, especially when used in the request prompts, e.g. talking head asks: "Are you sure that the correct answers is B birds?" and yields the turn to the user. The ECA architecture in a form of talking head is described in the following text.

3.4 Multimodal ECA-centered Application Platform

ECAs can communicate in several modality dimensions/channels (e.g. voice, gestures, emotions, non-verbal cues). The presentation of these channels needs to be synchronized to be able to mimic human-like behavior. ECAs can be employed as front-ends for dialog applications in some areas of use (e.g. as anchors in games).

However, when a developer needs to build truly conversational agent system, ECA is just one part of a complex system. The ECA provides means of output presentation layer. Spoken dialog applications mostly interact with a user through GUI and speech. This can be replaced by ECA, of course with changes to the dialog manager system that needs to control new communication channels introduced by ECA.

To build a fully-fledged ECA platform and save the time needed to develop our own spoken dialog system we, needed an existing dialog platform. The Conversational Interaction Manager Architecture (CIMA) platform was selected [149].

3.4.1 Dialog Manager

The CIMA is a dialog management framework that provides programming means and tools for authoring conversational applications both in speech-only and multimodal domains. Customized dialog management is achieved via the concept of dialog strategies implemented through dialog templates which are customized to provided application-specific call flows and dialog behavior. The programming paradigm is event-driven and combines declarative and procedural approaches to application development. Applications provided in declarative form include maps of dialog states, expressed in SCXML (State Chart XML, `http://www.w3.org/TR/scxml`) and resources that define (fragments of) grammars, spoken prompts and pieces of text or graphics to be shown visually. Procedural application logic is coded in JavaScript (Mozilla SpiderMonkey implementation, `http://www.mozilla.org/js/spidermonkey` or natively in C/C++.

To integrate multiple components that comprise a particular application, CIMA acts as a hub that facilitates event-based communication among the components and between the components and running applications. Standard components include speech recognition, synthesis, grammar generation, GUI and infrastructural components such as debugger console.

Key features that CIMA supports include:

- State-of-the-art statistical language modeling and natural language understanding technology.

- Free-form recognition of multiple items in a single request (e.g. "I want to listen to Walk by Foo Fighters").

- Free-form recognition of items from very large lists – for example, thousands of cities in a country or user's contact list. Users can simply say sentences like, "I need directions to 1600 Pennsylvania Avenue" without having to choose a State or explicitly getting into a city selection dialog.

- Recognition and disambiguation of partial names - for example, users can say partial names of streets, artists, etc., without having to remember official names.

High-level CIMA architecture is depicted on Figure 3.6. The CIMA dialog manager is used as a basic building block of the presented ECA-based architecture in the next section.

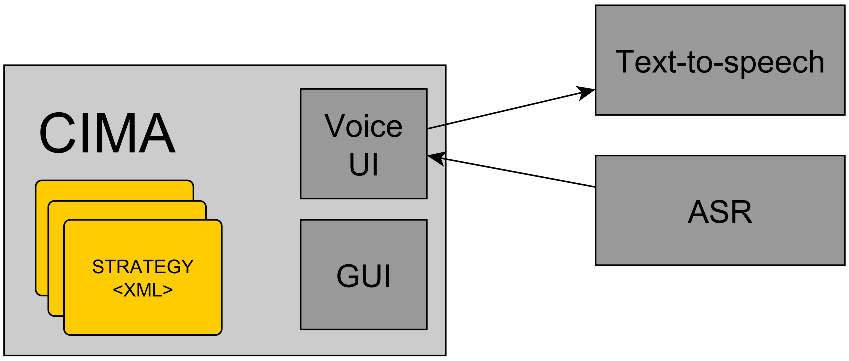


Figure 3.6: High-level overview of CIMA interaction manager architecture

3.4.2 ECA architecture

The basic concept of the architecture design of the ECA framework is the client-server communication paradigm (see Figure 3.7). The server listens on a specific network port and receives commands through a bi-directional communication link, mainly exercising ECA authoring language which is described in the next chapter. This allows to control the ECA behavior from a client in real-time. The ChiliX [42] library is used for transporting commands through TCP/IP network. The server also allows to use plain TCP connection to receive ECA language commands.

We use talking head ECA only. For some scenarios it would be helpful to use whole body agent (e.g. hand gestures, postures). However, sometimes head-only model can be more convenient, especially when combining ECA interface with "classical" interfaces. The head-only model allows much more freedom of movement and leaves larger space on the display for the rest of the user interface. When the application developers need pointing gestures, they can use various forms of "virtual" pointers (e.g. arrows, hand icons) that can be animated in the user interface.

The heart of the ECA server is the open-source Expression toolkit [61]. This toolkit renders and controls the 3D head model. It allows to include significant modifications that this thesis proposes. It is a procedural face animation toolkit which displays a 3D head model using computer accelerated OpenGL system. Movements of the face are simulated by a model of human muscles. These muscles are controlled by parameters in the Expression toolkit.

Since speech is one of the ECA communication channel, support for text-to-speech systems is incorporated. Both embedded and server version of IBM eViaVoice and IBM

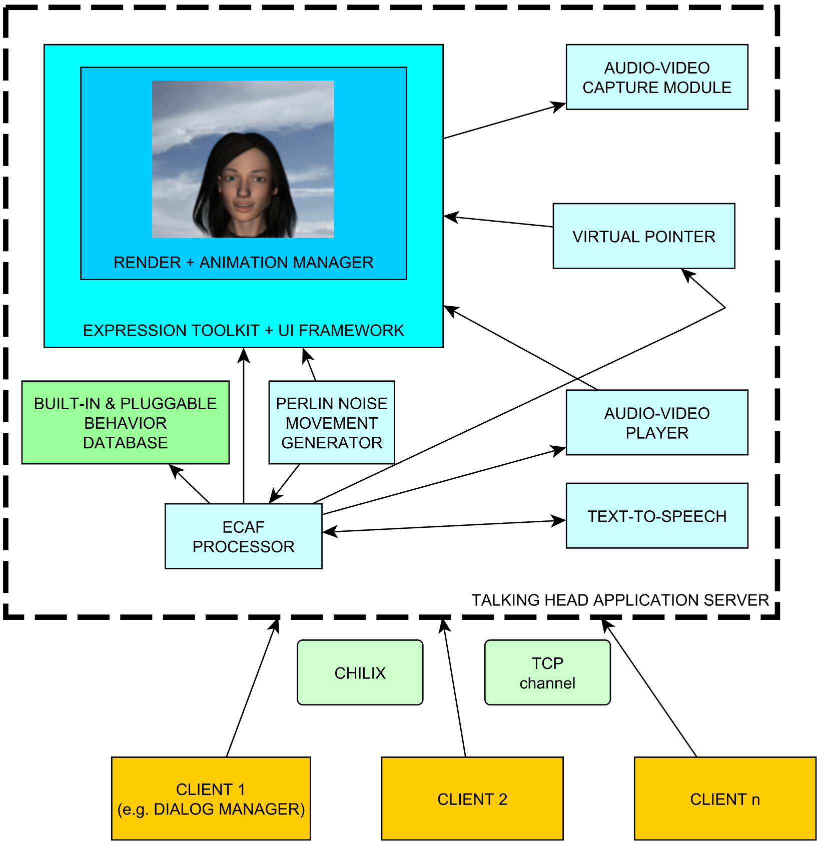


Figure 3.7: High-level overview of ECA architecture

Server TTS are supported [82]. This synthesizers represent state-of-the-art text-to-speech synthesizers that use concatenative synthesis. The embedded synthesizer is designed for small systems, the second one can be used on powerful PCs because of memory and CPU requirements and provides a better sounding voice. Supporting both synthesizers enables system architects to design ECA applications for various environments.

The tools introduced in the previous text are basic blocks to build ECA-based dialog applications. As it was mentioned in the Chapter 2 some typical problems arise when developing ECA applications. The following sections explain these problems and sketch solutions.

3.4.3 ECA Authoring Language

Most of the ECA languages and toolkits tend to split the application authoring task into (a) off-line step for preprocessing and (b) the real-time step of running preprocessed animations. This comes with the implicit disadvantage, that the animations are realized on a closed set of precomputed behaviors. Moreover, the authors seem to separate the process of designing the application to the design of ECA and design of user interface around ECA. To deliver the best usable notion of interaction these steps need to be combined. All “communication channels” need to be synchronized.

To address the above problems, the Embodied Conversational Agent Facade (ECAF) Language, scripting and controlling language for authoring applications based on avatars

Table 3.2: Four evaluated behavioral parameters of ECA (default value means ECAF toolkit default settings)

ECA Parameter	**Evaluated values**
eye's blinking	no blinking, default ($p = 0.0085$), fast blinking ($p = 0.1$)
teeth color (static)	pure white (default), grey-yellow, darker grey-yellow
mouth opening (speech)	less (80%), default (100%), more (110%)
head movements (speech)	no movement, default, faster movements

is proposed. The ECAF needs to be designed with the user-centered design approach, where most of the features are responding to direct needs of developers. This markup-based language is designed to meet the following requirements:

- fast learning curve,
- real-time performance,
- built-in extensibility to support future ECA functionalities,
- ability to control the user interface and ECA at the same time

The language should be based on the principle of mixing various interaction channels as most of the successful ECA authoring techniques utilize this paradigm. Moreover, this design pattern should allow future extension of supported channels.

The authoring language allows to setup various parameters of ECA behavior (e.g. head view angle, face expression, head position in the application space, size of head). An important part of ECA behavior is its appearance.

3.4.4 Measuring ECA Appearance Parameters

Implementing characters with believable human behavior is rather expensive and long task. Agent's visual apparance can be viewed from static and dynamic point of view. *Static behavior* is set by agent's character designer through selecting, e.g. color of eyes, hair style, color of hair, size of nose. Interacting agent express also *dynamic behavior*. Setting dynamic behavior of the agent incudes e.g. mouth opening, head movements, eye blinking. Proper setting of the both "static" and "dynamic" parameters can influence the real interaction with the agent and can lead to more natural and enjoyable ECA.

This section proposes evaluation of the set of ECA behavior (appearance) parameters. We would like to evaluate sensitivity of human users to these appearance parameters. Our hypothesis is that some "dynamic" parameters are more important for the user than the other ones. We have chosen and investigated four possible behavior (appearance) parameters that could be changed on the ECA and evaluate them in a user study. The parameters are listed in Table 3.2.

We selected four behavior (appearance) parameters based on visibility and users comments. The values of selected parameters can be easily changed in the ECAF toolkit. First three parameters are intuitive, but the last one needs more explanation. Humans are not static all the time. They move their bodies and heads continuously. Therefore, the last parameter "head movement" explore these involuntary movements using the ECA.

Humans are sensitive to perception of face muscle movements and expressions that the other human sends. A realistic visual articulation of an ECA is of high importance. The following section describes our contribution to one of possible tests of articulation.

3.4.5 McGurk Articulation Test Improvement

ECA visual articulation is a complex problem. Judging overall articulation is a very subjective matter. We need some human-like effects that influence our articulation that can be assessed on ECAs. The McGurk effect test is a method to evaluate speech articulation. The McGurk effect shows that humans use both hearing and vision modalities in parallel to perceive and understand speech. The first experiment was presented in [115]; there was a dubbed videotape of visual *ga* syllable with audio *ba* syllable. Experiment's participants thought that *da* syllable was pronounced.

The McGurk effect is used as one of the articulation evaluation tests to compare the quality of ECA's speech visualization. Masaro and Cosker used this kind of evaluation in their works [110], [33]. The test measures confusion responses on given McGurk video sequences with ECA. Authors of tests predominatly describe the group of participants as humans with normal hearing and vision. Often it is not mentioned whether participants had corrected vision or not.

Poor visual acuity can lead to changes in brain and to a decrease of neuronal activity due to reduced visual stimulation [197]. Thus, corrected vision can be the source of visual perception differences. Our hypothesis is the following:

People with corrected vision will judge the McGurk effect differently than people with non-corrected normal vision

The proposed experiment that uses a tape-recorded human and ECA should validate/invalidate our hypothesis. This section described predominantly basic parameters of ECA. The settings of these parameters is very important and will be discussed and evaluated in Section 5.3 and 5.1. However, the following section will try to address more complex problems related to the combination of ECA+UI behavior.

3.5 The Seamless Combination of Classical UI with ECA

Typical ECA authoring languages and toolkits separate ECA authoring from the UI environment authoring. However, the design process of multimodal applications involves both parts to synchronously allow consistent interaction with a user.

We propose an authoring language that should handle both parts of the world. An ECA discloses many new communication channels (e.g. expressions, behavior, movement on the screen, etc.). These channels should be accessible to the designer of the ECA applications. Classical UI provides means of interaction like text labels, images, animations/videos, buttons, etc.

Humans follow some interaction patterns when communicating with other people. Some of the communication patterns can be supported via an ECA. To achieve realistic dynamic appearance of the ECA, it should display some random movements, because humans are not static. Another example of interaction pattern is the way a hearer follows a conversation. People tend to turn the head or eye sight to the speaker. The mobile versions of ECA applications are more suitable than desktop or kiosk version there, because of the closeness of the user.

The environmental context of an ECA application use is an important characteristic that needs to be taken into account. Each context of use has specific requirements/needs when designing or using an ECA application. Main contexts of use are the following (sorted according to the closeness of the user):

- Mobile environment
- Personal computer environment
- Tablet environment
- Virtual-reality environment (Kiosk-based environment)

Mobile environment. Mobile or smart phones represent the environments that are the closest ones to the user. Smart phones provide a small display space. The expressions can sometimes be misunderstood on these devices, because smart phones can be used as the user walks or rides mass transportation, in a busy environment. On the other hand, closed office or home environment provides a relatively good level of privacy. Another character of this environment is relatively low CPU/GPU power for rendering an ECA and for synthesizing speech. This has much improved nowadays but the speed of animation is important from the battery point of view. Sophisticated ECA animation algorithms consume much needed battery power. This is an area to simplify algorithms for ECA animations and rendering.

One possibility to accelerate that process is to use coarser 3D model of ECA. If the head model has small amount of vertices and faces the task of animating such head will be easier. For example, Aubel tried to use impostors to increase frame rate of animation [9]. They presented technique that improves the display rate of animated characters by acting on the sole geometric and rendering information.

An alternative approach is the reduction of head dynamics. This is the method of CPU-power reduction which is proposed in this thesis. Our ECA model is animated using pseudo-muscles controlled by parameters. Vertices in the zone of muscle's influence are deformed during animation. The computation of this translation costs CPU/GPU power. The hypothesis is following:

Hypothesis H3 – *If the number of muscles defined in the ECA is reduced the amount of deformed vertices is reduced too; the computational cost of the ECA animation will be lower. This approach also reduces the detail of animation.*

Personal computer environment. Personal computers tend to provide more space between ECA and user in comparison to the previous environment of a mobile device. If an external display is used the distance can be even larger. The personal computers provide relatively high CPU and GPU power for the ECA, thus the rendering of the ECA can be detailed. However, the method sketched in the previous section can be used here when the CPU demands need to be lowered, e.g. when the computer is used for other tasks.

External displays offer a large space for ECA applications. The need of synchronizing with classical user interface is more apparent here. The expressive power of an authoring language should provide a large palette of UI components that can be integrated with the ECA presentation. In the terms of ECA behavior the user can interact with ECA from various angles of view. The ECA should follow the user by rotating and tilting its head or body. We should address that in this environmental context offering such behavior pattern to ECA application developers.

Tablet environment. This environment fills the market gap between the environments of personal computers and mobile devices. Tablet devices give the user freedom of movement. It provides the ECA application designer with relatively good CPU/GPU power but battery draining is still an issue here. Thus, adaptive dynamic rendering of animations will be useful here too.

This environment provides means of "touch interface". The interface is known for that it is very intuitive. However, it brings some shortcomings of that the user's hands can be sometimes very shaky (in the public transit environments, or caused by changes in motoric systems of older users). The user interface should be prepared for that and the buttons should be big enough and specific thresholds of touch gestures need to be counted in. This is more needed when working with older adults because their gestures can be relatively noisy.

Kiosk-based environment While the previous environments are situated mostly in private space, kiosk-based installations of ECA applications are meant for public spaces, e.g. hallways in offices, schools, stations, etc. Assumptions about a user can be made for one session only. The next user can be somebody else. From the space point of view these environments provide the biggest interaction space between the user and the ECA.

Speech recognition models should be prepared for noisy environments and preferably long-range microphones should be used. Microphone arrays and some way of user's tracking can be used. For example, Microsoft Kinect is a suitable and low cost device that can stand up for both tasks. Face or body tracking can also be used by ECAs to follow its users by eye-sight. The CPU/GPU power is the same as in case of personal computer based environments.

The interaction can be performed using hand or body gestures that are recognized by the means of computer vision, or touch screens can be used for closer interaction with virtual-reality based environments.

Environmental context of an ECA application use can be exploited when selecting various behavior patterns (e.g. size of head, position of head, face expressions, turn-yielding cue, telling private information or not). Fine-grained setting of these patterns is very tedious for application developers. The following section describes a high-level extension of an authoring language that can facilitate the role of an ECA application developer.

3.5.1 Ontology-based Contextual Behavior Patterns

The previous paragraphs described the contexts of ECA applications in terms of environment where the ECA applications interact with their users. However, this is not the only one parameter of "ECA context". Further, parameters like user's age, experience with computer programs, nature of the user, etc. are important to achieve the most natural interaction with user.

The character of presented information is the next interesting component of ECA context. For example, reading part of a news article can be associated with one high-level behavior pattern but reading a good joke can use another.

It is natural to help the ECA application designer to provide some way of controlling which behavior patterns can be used in a specific environment of interaction and based on what an ECA is presenting.

We propose a knowledge base that contains various behavior patterns for specific contexts of ECA use. This knowledge base will have an ontologically defined structure. The generality of ontologies allows us to design a structure that can be further extended if needed. Selected behavior patterns can be plugged in or taken away.

The ontology will define ECA behavior patterns dependent on context. The knowledge base can also contain the user context that can be gathered from the user using three ways:

1. *Manually* – The information about a user is collected by system designer or operator and predefined parameters like age or user abilities are set before the system is used for the first time.

2. *Automatically* – During the usage of an ECA application the system collects the data, for example to re-estimate the thresholds used to recognize touch gestures over buttons.

3. *Semi-automatically* – The parameters are set by the dialog manager that uses an ECA application as a front-end (e.g. preferred talking head background).

The ontological modeling can also help the authoring language to be interoperable. For example, the setting of face expression can be defined by textual description e.g. ”smile”, ”happy”, ”sad”, etc. or using parameters of face muscles. The ontology can define many possibilities of a face expression description. Thus, this gives the ECA application designer the freedom of choice how detailed the description of the gesture should be.

3.6 Summary

The chapter has described some issues that exist in the world of humanized ECA interfaces. The proposed solutions will be tested and used when developing the serious game “Who wants to be Billionaire”. The game, where ECA works as moderator. The principle is known from the famous TV show. The user there answers questions asked by the moderator. The game will be interacting multimodally, using speech recognition, touch-based gestures, mouse or keyboard.

Spoken dialog systems prevalently use push-to-speak buttons or wake-up words to start speech recognition. This is not very natural, because human-to-human conversation are not interrupted by anything similar to buttons. The research of turn-taking and turn-yielding tries to change this behavior and to use some natural cues that people use. The area of turn-taking and turn-yielding is explored mainly from the spoken dialog systems point of view. The research of turn-yielding cues focuses on vocal ones mainly. One goal of the dissertation thesis is to explore the possibilities of facial turn-yielding cues in ECA interfaces.

There are already some models of turn-taking presented by researches. However, these models take into account mainly vocal turn-yielding cues. Not much has been done in the world of ECA facial turn-yielding cues. We propose new visual turn-yielding cues that will be evaluated in an ECA application. The visual turn-yielding cues evaluated here are following:

- Stopped talking head movement at the end of utterance
- Talking head nod at the end of utterance

Further, one would like to know whether the number of cues helps the ECA to better convey the turn-yielding signal. So, the following hypothesis needs to be evaluated:

Hypothesis H2 – *Using more turn-yielding cues before a transition relevance place increases the probability of the correct judgment about the next speaker. The turn-yielding cues can be both vocal and visual.*

The second class of problems in the ECA field is the way of combining "classical user interfaces" and the ECA user interface. Good features from both of the worlds are combined. This can be done by designing an authoring language that provides the application designers with freedom to decide whether they need to access low-level functions or high-level concepts when defining the ECA behavior.

Use of ontologies is proposed to define the structure of the knowledge base which contains the ECA behavior patterns. These patterns are dependent on the context of ECA application use. The context of ECA application is mostly influenced by the environment. There are several environments the ECA can be running in. The main contexts of use are the following:

- Mobile environment
- Personal computer environment
- Tablet environment
- Kiosk-based environment

However, an environment context is not the only context that involves interaction with an ECA. Other parameters like user's age, experience with computer programs, nature of a user influence the interaction. From the ECA point of view, a topic which ECA presents influences the behavior pattern (e.g. news, jokes or games, etc.). To make the switching of behavior patterns easier, a context dependent ontologically modeled knowledge base is proposed. The new patterns can be easily plugged into the knowledge base and used by various ECA applications. The next chapter describes the realization of the proposed solutions and sketches evaluation procedures.

4 Realization

This chapter describes solutions that are mainly used to build the "Who Wants to Be A Billionaire" ECA-based game. The target group for this game consists of active older people that want to live independent life as long as possible. This game should train their cognitive and brain functions.

To build a successful system, the whole process of system implementation should be user-centered and the application should be implemented with the knowledge of our target group. The personas paradigm can help us during the whole development of our application.

4.1 User specification

The primary target users for the Netcarity project are independent older people living alone. There are several assumptions and categorization of key characteristic factors of our target groups. These come from the interviews with the representatives of our target groups. The key characteristics are the following:

- Does not want to be home alone.
- Does not like long traveling.
- Does not understand the modern technologies including mobile phone, computer, etc.
- Does not want to learn new things.
- Often travels to visit children or friends.
- Follows current news and events.
- Prefers to rest in quiet environments.
- Age from 60 to 90 years.

The characteristic features in which some representative users differ are the following:

- Has computer and uses it / Does not have computer
- Watches television more / reads newspapers or books more
- Plays with grandchildren
- Has children and lives with them / Lives alone / Lives in a senior center
- Has a pet / Does not have a pet
- Has mobile phone / Does not have mobile phone
- Wants to learn new technologies / Does not want to learn anything new. (scared of technologies)

Figure 4.1: Primary persona Anne

- Wants to stay independent
- Prefers social activities / Prefers to stay alone
- Disabilities (eye regressions, shaking of hands – regressions in body motorics)

These key characteristics can help us construct the persona. This persona should be a typical representative of our target user group.

4.1.1 Primary persona Anne

Anne is a 70 years old vital woman. She is depicted in Figure 4.1. She worked as a cook in school cafeteria. Her husband died six years ago. She is living in her own flat with a dog called Brit. She has two children, two sons. She has six grandchildren. She likes to visit her son's families and likes to host their visits in her flat on special occasions like birthday or Easter.

Anne tries to live actively. She watches current events in politics, reads newspaper and listen to news on TV. Anne enjoys smart games. She likes to watch popular TV shows with factual questions. She also likes to do crosswords. Every Friday she goes to a senior center to play cards and to chat with her friends.

Anne owns television and uses teletext to get information about weather. She does not have a computer but her son taught her to use his computer to write emails. Anne has a mobile phone too. She is able to receive and place calls and to send short messages. However, she is struggling with small font in her mobile phone, so she likes to place a call instead. She wants to stay independent as long as possible.

The persona of Anne will be driving the development of the user interaction part of the Billionaire game application. However, basic building blocks are needed to develop the speech-enabled Billionaire game. One crucial part of this development would be the ECA technology.

4.2 ECAF Talking Head Toolkit Architecture

There are various domains where ECAs could be these agents helpful. One example is using them as a narrator of multimodal applications or presentations to make them more dynamic and trustful. Secondly agents can be used as a natural computer interface when dealing with older people. They are more appealing for them than "cold" old-fashioned standard interfaces.

Lip-synchronized talking head is an example of such an agent. Facial animation is a challenging task in computer graphics. There are many ways of simulation of the human head. The first simulation of faces was done by Parke by interpolating positions of vertices between extreme positions [139]. The interpolation of vertices positions is not very effective so the parametric model appeared [140]. Extended methods use physics-based muscle modeling and simulation. There are three variants: spring mesh muscle [146], vector muscles [190], and layered spring mesh muscles [176].

In the ECAF Talking Head toolkit, a pseudo-muscle model of facial animation is used as an internal graphical representation of an agent [190].

The basic concept of the architecture design of the ECAF framework is the client-server communication paradigm (see Fig. 3.7, in Section 3.4.2). The server listens on a specific network port and receives commands through a bi-directional communication link, mainly exercising two ECAF commands – ACT and SPEAK, which we describe in more detail in Section 4.3. The client controls the behavior of the avatar by connecting to the server and sending a stream of ACT and SPEAK commands performed in real-time. In our case we use the ChiliX [186] library for transporting commands through TCP/IP network.

At the heart of the talking head server is the open-source Expression toolkit [62], which has been significantly modified to match it up with the ECAF language capabilities. This toolkit renders and controls the 3D head model. It displays the head model with aid of the OpenGL system [198]. Movements of the face are simulated by a model of human muscles. These muscles are controlled by parameters in the Expression toolkit. We used a 3D model of woman head that is called Masha. The license to the 3D model was acquired [96].

4.2.1 Used Facial Animation Methods

The simplest facial animation method is called interpolation of key-frames [139]. Separate 3D meshes must be defined for every expression which the system is able to display. Further, the animation is only time interpolation between 3D vertex positions. This method has one big disadvantage that the animator must prepare meshes for all the expressions manually.

This problem solves a parametric facial animation system. One of the most known systems is pseudo-muscle deformation model [190].

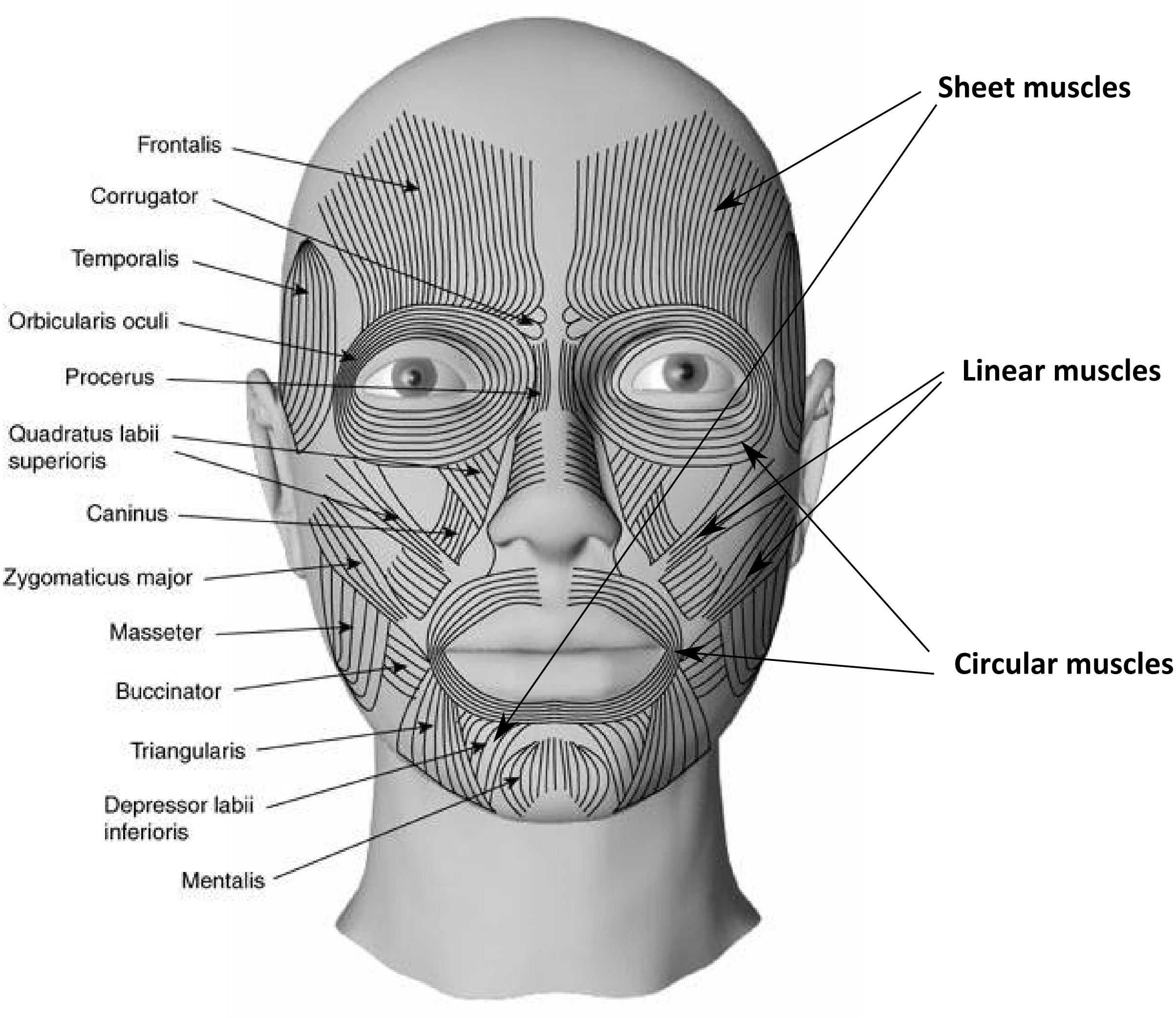


Figure 4.2: Human muscles (Picture courtesy of P. Ratner)

4.2.2 Pseudo-muscle Deformation Model

This model is inspired by the human head anatomy. The human head consists of the skull, muscles and skin. This simulation model works on the basis of contracting and relaxing the expression muscles on the head. Every possible expression of the head corresponds to a set of parameters that controls contraction of each muscle.

The real expression muscles are divided into three groups: linear muscles, sheet muscles and circular muscles. See Figure 4.2.

The extended pseudo-muscle model precisely copies this division into three groups. Each muscle is approximated by the space (cone, sphere, etc.) of influence to the mesh vertices; it controls a definite amount of vertices. The contraction of muscle is expressed by a real (or integer) number. Greater number expresses greater contraction of the muscle.

Figure 4.3 illustrates an example visualization of pseudo-muscles. Red dots are vertices that are influenced by the muscle.

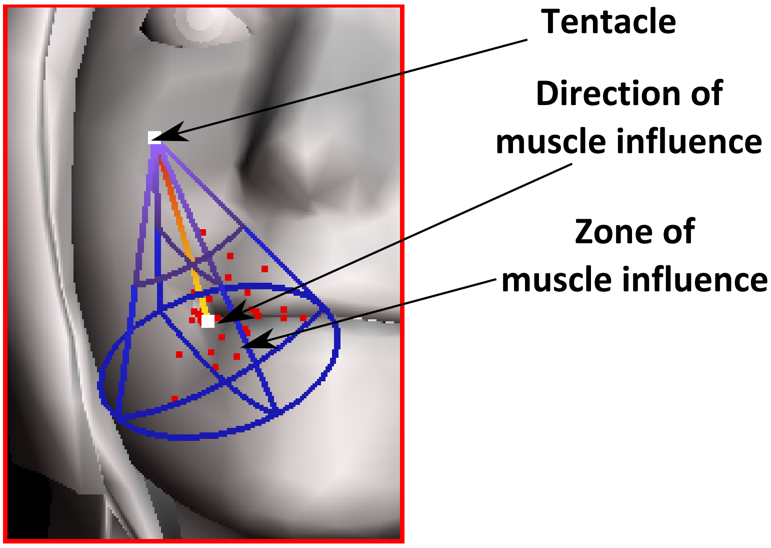


Figure 4.3: Linear pseudo-muscle

During the animation, muscle parameters are interpolated between the expressions. This makes the animation smoother in opposite to vertices interpolation method.

4.2.3 Lip synchronization

Since the ECAF avatar supports lip-synchronization, we incorporated support for the IBM eViaVoice Text-To-Speech engine [49]. The synthesizer outputs various speech parameters during synthesis, from which the phoneme sequences are used. These sequences are translated to visemes used for lip synchronization.

4.2.4 Random Movements of the Head

To achieve realistic dynamic appearance of the avatar, random movements of the head were implemented. These movements are generated by 1-dimensional Perlin noise generator [141]. Perlin generator is a function that adds controllable pseudo-random noise to the avatar head's movement. Based on user feedback, we tuned the Perlin noise generator to obtain pretty realistic appearance of the head. The level of Perlin noise is a configurable parameter that can be set by the application designer.

4.3 ECAF Authoring Language

In this section, the ECAF Language is presented as a scripting and controlling language for authoring applications based on synthetic avatars (Talking Heads). The ECAF is designed with developers needs design approach, where most of the features correspond to direct needs of developers. It was intended to design a markup based language with a fast learning curve, real-time performance, and built-in extensibility to support future avatar functionalities.

Real time Behavior Control. As already indicated, the talking head could be controlled through two basic control elements – the ACT and SPEAK. With the aid of

Table 4.1: Communication channels supported by ECAF

Meta channels	**Communication channel**	**Affected by**
speech	voice	`<speak>`
visual	head turning	`<gesture head_angle="">`
	eyes pointing	`<gesture eye_horiz="" eye_vert="">`
	facial expression	`<gesture expr="" expr_scale="">`
	background picture	`<gesture background="">`
	text window	`<text>` and `<gesture text="">`
	head size	`<gesture zoom="">`
	head position	`<gesture head_x="" head_y="">`
	body posture	`<gesture posture="">`
	overlay pictures	`<overlay>`
	video overlay	`<media_play>, <media_stop>`
	head zoom	`<gesture zoom=">`
	animation time	`<gesture move_time="">`
	virtual pointer	`<gesture pointer="">`

the ACT command, we immediately change the appearance and behavior of the Talking Head. For example, the application developer can send a command to turn head 20 degrees right, which results in real-time response showing the animation of turning head.

The SPEAK command is closely connected to the speech synthesizer. It prescribes the utterance which is sent to the synthesizer and the head shows individual visemes[1] as this utterance is synthesized and played through a sound card. By these two high-level commands we are able to cover many of the communication channels supported by ECAF as depicted in Table 4.1.

4.3.1 The ACT command

The ACT command is realized by the XML `<act>` element. The ACT command has only *one* XML element child. Action carried out by this command is specified by one of these elements: `<text>`, `<stop>` `<start_capture>`, `<stop_capture>`, `<gesture>`.

`<text>` element. As the talking head is a multimodal application, we need support for showing textual output. Plain text to display is the only child of the `<text>` element. The toolkit displays a rectangle window near the head and writes the text into this window. New line is marked by the *$* symbol. For example, displaying the text: *”Traffic jam ↩ Highway A6”*, is realized by issuing the command: `<act><text>Traffic jam$Highway A6</text></act>`.

Plain text display can also be controlled by an API that allows to setup more extensive properties of text. It introduces referencing pointers to manipulate multiple texts in real time. The `text` is extended by these parameters:

1. `ref` – integer reference value. It works as a pointer while manipulating the text instance

2. `pos_x` – x-coordinate of the text base point in percentage of the window width.

[1] The approximation of lips and face shape that corresponds to the phoneme.

3. `pos_y` – y-coordinate of the text base point in percentage of the window height.

4. `height` – font height in pixels.

5. `wrap` – turns-on automatic wrapping of the text. Maximal number of characters on text-line.

6. `font` – path to the TrueType font file in the file system

7. `delete` – deletes referenced instance of text.

`<stop>` element. When this command is received by the talking head, it tries to immediately stop the synthesis[2]. The head will also return into the neutral position and display the default expression.

`<start_capture>` and `<stop_capture>` elements are used for capturing video and audio track of the real-time animations for debugging and archiving purposes.

`<gesture>` element is the most complex element in our language. It controls several talking head channels including movements and expressions. These channels are affected by attributes of this element and by their values. As there are nine independent communication channels supported by the ECAF talking head, the gesture element can contain these attributes:

1. *head_angle* – Integer attribute value which represents the angle of a head turning in the horizontal plane.

2. *eye_horiz* – Integer attribute value which drives the talking head eyes turning in the horizontal plane.

3. *eye_vert* – This attribute which has the same meaning as parameter *eye_vert*, but in the vertical plane.

4. *expr* – A value of the *expr* parameter is a string from the table of supported face expressions. The head will show this expression. Expression strings are for example: neutral, smile, anger, ...

5. *expr_scale* – A floating point number that depicts a scale of an expression. Number 1.0 represents full expression, while 0.0 displays no expression. The parameter *expr_scale* must be used only in combination with the *expr* parameter.

6. *idle* – A string from the table of the idle movements that turns on the idle movement of the head. For example, *FlyOut* means that the talking head will start an animation of the head disappearing in the perspective.

7. *background* – The attribute contains the name of the image file to be loaded as the window background.

8. *text* – Text child has the same meaning as the `<text>` element.

[2] Immediately means in the case of IBM eViaVoice synthesizer after the next word.

9. *zoom* – This attribute takes in a floating point number that denotes the zoom of the head. This allows to make the head bigger or smaller depending on the application scenario.

10. *head_x* and *head_y* – These parameters control the position of the head in the window. They allow to move the head over the background image onto the desired coordinates depending on the application scenario.

```
<ecaf>
  <speak src="helloworld.wav">
    <phoneme time="0.611">E</phoneme>
    <phoneme time="0.787">l</phoneme>
    <phoneme time="0.845">o</phoneme>
    <gesture head_angle="15">
      <phoneme time="0.925">w</phoneme>
      <phoneme time="1.06">r</phoneme>
      <phoneme time="1.170">l</phoneme>
      <phoneme time="1.252">d</phoneme>
    </gesture>
  </speak>
</ecaf>
```

Example 6: Speech recorded in a sound file scenario example

The SPEAK command. The SPEAK command is encoded by the tags `<speak>` and `</speak>`. Its content is the text to be spoken. This text is sent to the speech synthesizer which produces the artificial speech delivered in realtime. The movements of avatar lips are synchronized with this speech. The SPEAK command can be combined with the `<gesture>` element (and all it's options) to modify the appearance of the head during the speech.

There is an alternate way how the developers can utilize the SPEAK channel. In case they need to lip-synchronize with a prerecorded sound (such as .wav file), the `<speak>` element will contain the name of a file as the attribute. The utterance will be represented by phoneme elements with a value of the phoneme including the time stamp of the beginning of this phoneme. You can see small illustration in Example 6.

Scenario Script. So far we have shown how the client application sends the SPEAK and ACT commands to the talking head server according to the application state. For testing purposes, and also for creation of static demo applications, it is possible to write the sequence of ECAF commands into an XML script file. Such a script is rooted within the `<ecaf>` and `</ecaf>` tags, which contain the sequence of `<speak>`, `<act>` and `<sleep>` tags. The `<sleep>` element is the execution command parameterized by a floating point number. The client program that reads scenario script file will stop sending next command for the time of seconds specified by the `<sleep>` element. The demonstration of the scenario files is presented in Examples 6 and 7.

The ECAF toolkit is the presentation part of the whole solution. There should be systems that can drive the dialog and provide the means of speech recognition. These functions are provided by the CIMA Framework that is described in the following section.

```
<ecaf>
  <speak>Hello everybody!</speak>
  <speak>How are you?<gesture expr="smile" expr_scale="0.8">
    I am fine.</gesture>Now I will turn eyes and head
    <gesture eye_vert="20"> I see the
    <gesture head_angle="-25" persistent="true" />top.</gesture>
  </speak>
  <speak><gesture head_angle="0" persistent="true" />I will turn
      head and <gesture expr="smile">
      give you a smile. <gesture head_angle="25">Look at me now!
      I am flexible.</gesture></gesture>
  </speak>
  <speak>I can show you, how I can <gesture expr="smile">mix
      expressions. <gesture expr="left blink">
      It's hard but it works</gesture></gesture>
  </speak>
  <act><gesture idle="LookAndSleep" persistent="true" /></act>
  <act><text>I'm sleeping$just now</text></act>
</ecaf>
```

Example 7: Text-To-Speech scenario example

4.4 ECAF CIMA Framework Architecture

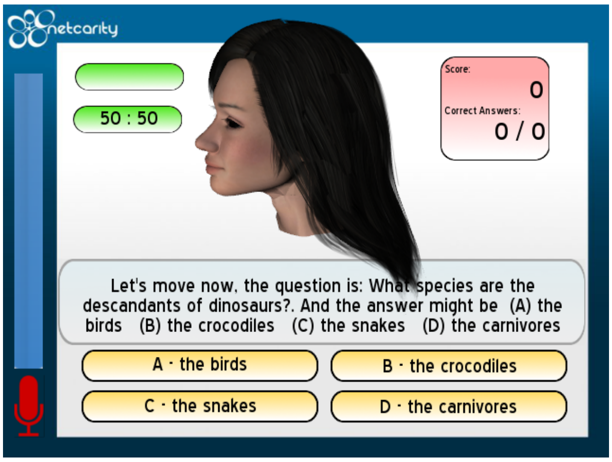
The Billionaire game is implemented as a system that consists of two distinct software components. One is the CIMA architecture which was briefly depicted in the previous chapter. The CIMA does the speech recognition and controls the flow of game. The CIMA communicates to the ECAF toolkit that takes care of the 3D Talking head anchor rendering and text prompt rendering to speech (see Figure 3.6).

The application code is mainly written in the JavaScript language, divided into two parts:

- GUI framework
- Voice framework

The former is responsible for the voice interface, the latter for the ECAF toolkit (talking head and GUI) events and interactions. CIMA allows to use SCXML program templates to define the behavior of dialog application. Instead of the SCXML templates, the Billionaire framework is built on basis of contexts (Context means one part of the communication represented by asked question and user's input). These contexts are switched using an internal application stack. Therefore on the contrary to SCXML templates, this approach allows the developer to effortlessly implement history functions and more.

At the beginning of each dialog iteration, the framework checks all contexts whether they are properly initialized. Then CIMA looks for all required grammars and registers them. Afterwards the prompt of the actual context is played and application waits in



Score: 0

Correct Answers: 0 / 0

Let's move now, the question is: What species are the descandants of dinosaurs?. And the answer might be (A) the birds (B) the crocodiles (C) the snakes (D) the carnivores

Figure 4.4: Talking looking at “HELP - 50:50” button while talking about help

the main loop for any events. At the end of processing the event, the application checks the state for the next iteration. There are some important types of contexts, see the list:

- *Prompt context* – This context defines text-to-speech prompt which is played. When the prompt is finished next state is entered.
- *Spoken input context* – Interactive context. Defines fully featured state that asks a question and then waits for user's input.

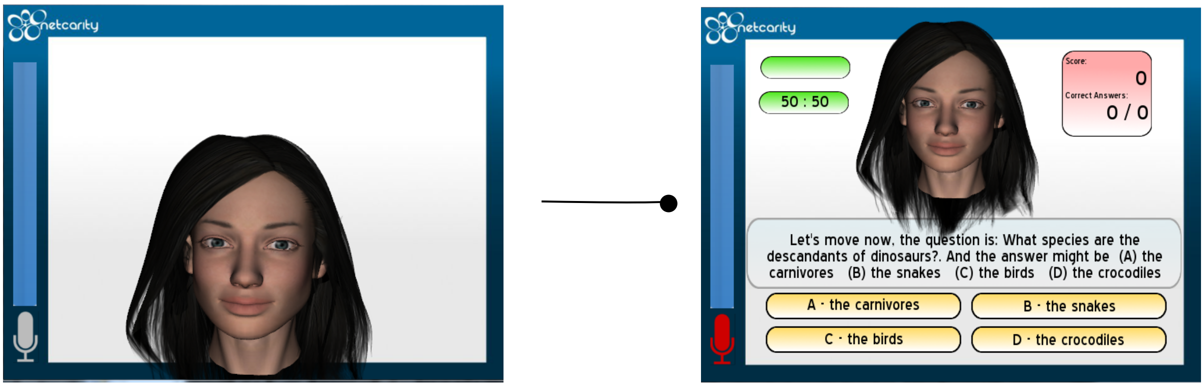
The overall framework (including JavaScript applications) also takes care of the following action commonly found in speech dialog applications.

4.4.1 Prompting

System prompts in a dialog system are important user indicators of what kind of user input is expected. E.g., a prompt “Please, tell me your answer A, B, C or D” inherently suggests a much narrower response domain than the opening prompt “Hi, what can I do for you?”. By using the ECA as the system front-end, it is believed to increase the bandwidth of possible system hints to the user in terms of what kind of input is expected by e.g. indicating ECA attention to a particular GUI item on the screen when speaking about it 4.4, varying the position and size of the ECA on the GUI canvas with respect to the presented visual and aural content 4.5, or using non-verbal features such as head shaking, smile, surprise gestures, or a specific hand position to indicate the expected response span for a given para-linguistic context that is thus communicated to the user 4.6. Our ECA is present all the time on the screen and so in a sense plays the role of the application moderator, so that at any instant the user always has a visual conversational anchor to which s/he can direct her/his requests and responses to system prompts.

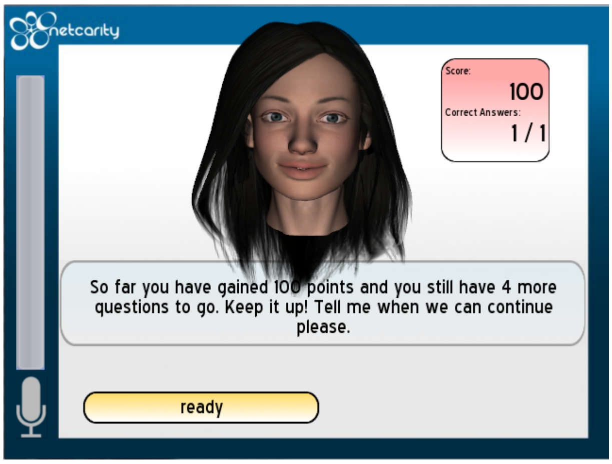
4.4.2 Disambiguation

Disambiguation is typically the most challenging part of any dialog application and good disambiguation capabilities differentiate successful dialog systems from less successful



Let's move now, the question is: What species are the descandants of dinosaurs?. And the answer might be (A) the carnivores (B) the snakes (C) the birds (D) the crocodiles

Figure 4.5: Changing the size of ECA according to the user interface space utilization



So far you have gained 100 points and you still have 4 more questions to go. Keep it up! Tell me when we can continue please.

Figure 4.6: Smiling talking head is telling the user that s/he successfully gained 100 points in the game.

ones. The goal of any dialog designer is to make disambiguation blocks intuitive by maximizing naturalness and by properly balancing (not necessarily minimizing) the number of man-machine turns. We experiment with the role of the ECA in the disambiguation process as a metaphor of a friend that helps the user make the right decision. The ECA is presenting the options to be disambiguated and works as a "tour guide" navigating the user through often complex disambiguation choices. In our case the ECA is looking at the answers together with the user and the talking head is reading and sometimes commenting the presented options. There is a broad range of possible conversation styles that can be used during disambiguation ranging from a plain presentation of choices using only basic visual gestures of head, eyes, and hand movements to a full infotainment experience where the ECA presents and comments the options in a very expressive form, appearing opinionated and assertive by adopting a style of some TV anchors. For the Billionaire game, we tried to stay in the middle of the spectrum – focusing more on the persuasiveness of the ECA as the disambiguation "buddy" without skewing her generated behavior toward excessively expressive emotions. The disambiguation process helps us determine the user's goal.

4.4.3 Action announcement

Once the goal of the conversation has been reached and the system understands what action needs to be carried out, it can perform the requested action. In the case of the Billionaire the action maps to evaluating the user's answer. The action can be performed right away or can have an overture, such as a prompt announcing the action, and for actions that cannot be undone also including confirmation. This system feedback can at one hand slow down the time-to-action; but on the other hand it may increase the level of robustness as well as naturalness of the application dialog logic. In our case, we experimented with the ECA announcements that consist of three sections: a) we confirm the user selection by repeating the full text of a question and answer; b) we experiment with reinforcing the user decision by generating ECA's positive comments such as "Good choice" with appropriate emotional expressions; and c) we convey additional information that may be helpful to the user for the current action context, e.g. "Are you sure that the selected answer is the right one"?

4.4.4 The single center of attention gravity

One key challenge was to resolve the problem of two centers of attention. From the previous project of ECA-based Jukebox we found that dividing the centers of attention in application is not good for users. When adding an ECA to the Jukebox application the initial user tests indicated that users have trouble to read the text on a multi-modal screen (the front-end of the existing Jukebox application) and at the same time to follow the ECA gestures including the lip, head, eye, and hand movements. For users this felt schizophrenic. We had to change the GUI front-end drastically, basically removing the old multi-modal GUI completely and focusing the new GUI around the ECA which represents the single center of attention. The Billionaire game user interface is therefore designed with the ECA as center of attention in mind. Next paragraphs describe the GUI design process in detail.

4.5 Billionaire GUI Evolution

This section describes the process of Billionaire GUI realization. The Billionaire game as other speech dialog application has specific demand on the graphical user interfaces because of the inherent application multimodality.

For implementing the GUI we enhanced our ECAF Talking head toolkit by CEGUI implementation [183]. The CEGUI is pretty general GUI framework that can be rendered to the OpenGL layer and it implements the needed features as event handling (e.g. when user clicks button) and it provides templates for advanced GUI components like progress bars. These can be used in speech dialog applications to show current energy level from microphone and in the case of Billionaire game they are used to represent the overall progress of game.

The first version of Billionaire GUI is depicted in Figure 3.4. This version of GUI has some deficiencies and is not very suitable for our target group and our Anne persona.

To design new Billionaire game GUI that is suitable for the target group of older people and that can be user tested. The new GUI is based on the Anne's requirements:

- Clear and easy control of game

- Clear graphics layout
- Chance to correct possible mistakes and errors
- Area with help

The whole GUI design was divided into four parts, concerning: color scheme, layout, user needs and requirements and system requirements:

- *Color scheme* – Blue "Netcarity logo" color was used as the basic color. The other colors symbolize different states of application. The red color symbolizes warning, green color means success and yellow is neutral.
- *Layout* – It has three basic parts: talking head, area for text information and area for complementary information.
- *User needs and requirements* – Because the goal is to make the game easily playable for older users, it is necessary to create whole graphic user interface simple and understandable. The game design uses contrast of colors for separation of each part of layout. The parts are big enough and because of using the color schema it has a clear purpose. The text should be also clearly readable.
- *System requirements* – The Billionaire game is developed for "tablet-like" devices. So, the screen is placed near to the user who controls the game flow using voice commands.

Figure 4.7 depicts the new prototype of GUI for the Billionaire game that is built under the conditions of requirements stated in the previous text.

The Billionaire game will be running on "tablet-like" devices which do not have enough power/or battery life. Therefore the talking head rendering level of detail needs to be discussed in the next section.

4.6 Talking Head Level of Detail

Trying to be very precise in the simulation of human head is a very computationally intensive problem, even on nowadays powerful computers. Certainly there are situations where one does not need to be precise, but instead the animation cost needs to be as small as possible. The small computers (PDAs, mobile phones, etc.) are the first example; they do not have enough power to do computationally expensive graphical tasks. Secondly if it is needed to run plenty of talking heads in one computer, one cannot be precise in all details and the application needs to simplify the process of facial animation.

One possibility to accelerate that process is to use a coarser 3D model of head. If the head model has small amount of vertices and faces, the task of animating such head will be easier. In this case we simplify the representation of the head model. Aubel tried to use impostors to increase frame rate of animation [9].

Two methods dealing with the level of detail are discussed in this section. One method is static and will be only briefly introduced. Its basis is in simplification of the 3D mesh. The second method we propose is based on reducing the number of deformed vertices.

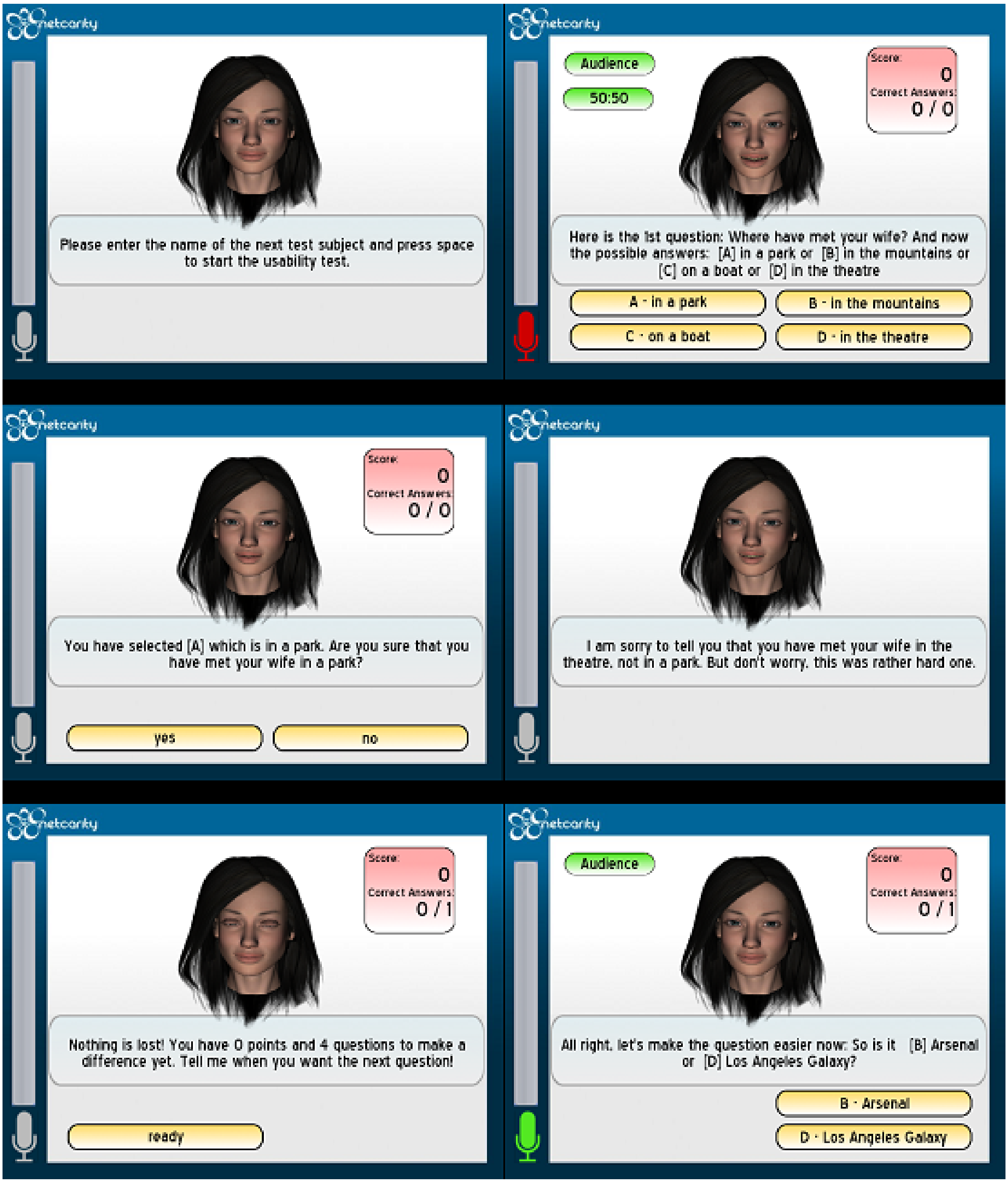

netcarity
Please enter the name of the next test subject and press space to start the usability test.
netcarity
Audience
50:50
Score:
0
Correct Answers:
0 / 0
Here is the 1st question: Where have met your wife? And now the possible answers: [A] in a park or [B] in the mountains or [C] on a boat or [D] in the theatre
A - in a park
B - in the mountains
C - on a boat
D - in the theatre
netcarity
Score:
0
Correct Answers:
0 / 0
You have selected [A] which is in a park. Are you sure that you have met your wife in a park?
yes
no
netcarity
I am sorry to tell you that you have met your wife in the theatre. not in a park. But don't worry. this was rather hard one.
netcarity
Score:
0
Correct Answers:
0 / 1
Nothing is lost! You have 0 points and 4 questions to make a difference yet. Tell me when you want the next question!
ready
netcarity
Audience
Score:
0
Correct Answers:
0 / 1
All right. let's make the question easier now: So is it [B] Arsenal or [D] Los Angeles Galaxy?
B - Arsenal
D - Los Angeles Galaxy

Figure 4.7: prototype of Billionaire game GUI for older people (walk through)

4.6.1 Adaptive 3D Mesh Simplification

A simple 3D mesh is rendered faster than a complex one. There are algorithms used in level of detail that simplify the mesh according to the external parameters (e.g. object size, distance from an observer), one in [199]. Adaptive refining of 3D face mesh is also elaborated in [206].

These methods include only simplifying the mesh adaptively according to the shape of the 3D mesh. Animating the talking head deforms the 3D mesh non-uniformly. Some parts of the 3D mesh are deformed more often then others. We suggest using the amount of deformation as a parameter of these mesh simplification methods. The most deformed parts of the 3D mesh will be simplified less than the non-deformed or less defformed ones.

The amount of mesh deformation is defined as Euclidean distance between the deformed and non-deformed vertex.

To get the mesh deformation statistics adapted to the application usage, one will need collection of typical words and expressions for that application and then it is needed to find out and normalize the amount of mesh deformation to the interval $\langle 0.0; 1.0 \rangle$. So each vertex of the mesh will be rated by a number from this interval.

Figure 4.8 shows the visualization of such measurement. This visualization uses mapping interval $\langle 0.0; 1.0 \rangle$ into so-called heat-map colour range (Figure 4.9).

4.6.2 Adaptive Reduction of Pseudo-muscles

Adaptive reduction of pseudo-muscles is based on a hypothesis that if we reduce the number of muscles defined on the head we also reduce the amount of deformed vertices; the computational cost of the head animation will be lower. Reduction of used muscles will also reduce the detail of animation.

This reduction of pseudo-muscles could be done adaptively according to a specific application. Each application that uses the talking head interface has a typical set of used words (animation of mouth) and expressions. As the animation of the talking head is running we could measure up the usage of every muscle. The animation of the head is an interpolation between two extreme key-frames. We will take every displayed expression parameters and from them we will find out the amount of each muscle contraction.

When normalizing these statistic data to the interval $\langle 0.0; 1.0 \rangle$ we can visualize them right on the head. The interval $\langle 0.0; 1.0 \rangle$ is transformed to colors as heat-map (see Figure 4.9). The linear and sheet muscles are visualized as colored strokes and circular muscles as 3D ellipsoids. Figure 4.10 denotes the situation.

This statistical information about pseudo-muscles we gain helps us adaptively reduce the details of the rendered talking head and reduce the computer or portable device workload.

When we need to reduce details we will start to turn off the muscles that have the lowest probability of usage according to our statistics. As it was said in upper paragraph this statistic is likely to be application specific; each application uses another portion of words and expressions.

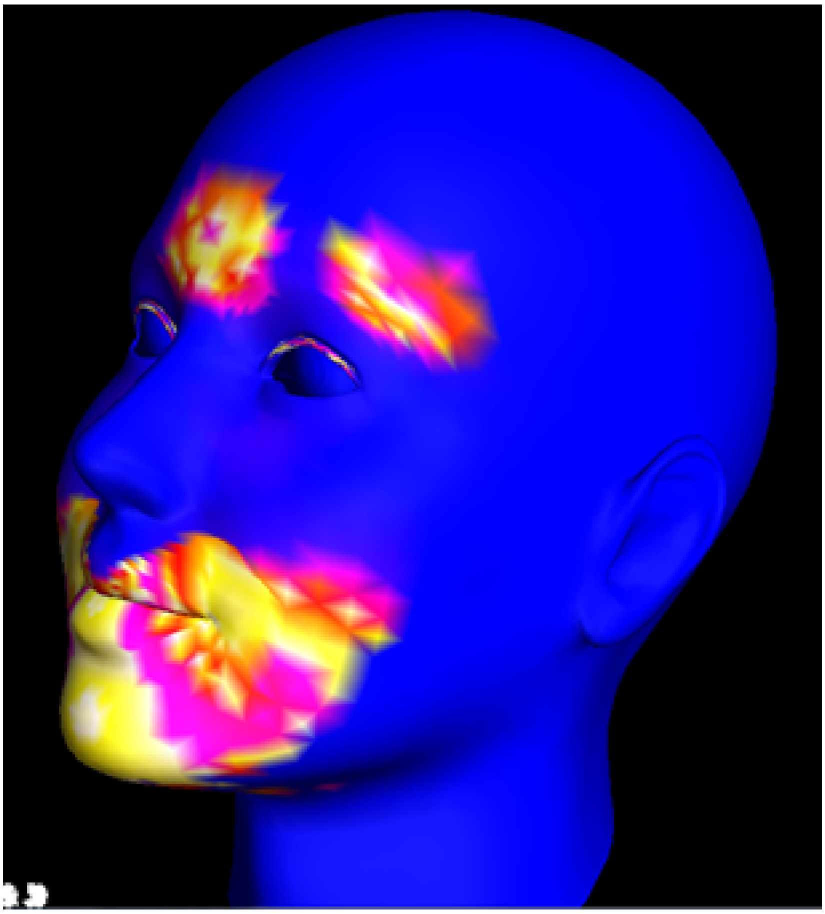


Figure 4.8: Visualization of the mesh deformation; Areas surrounding the mouth and the area of forehead are the most deformed ones.

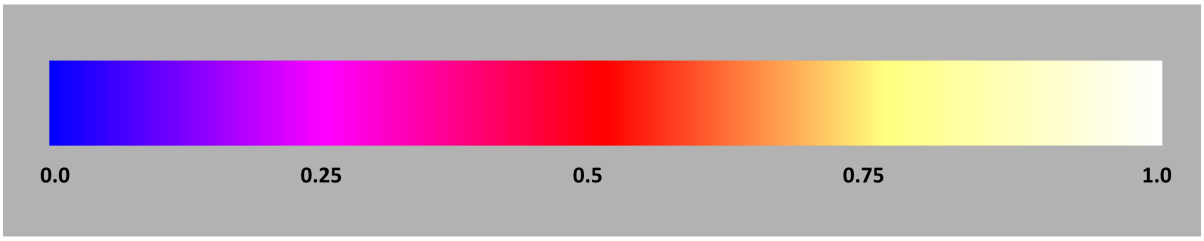


Figure 4.9: Transformation of $< 0.0; 1.0 >$ interval to colors (heat-map)

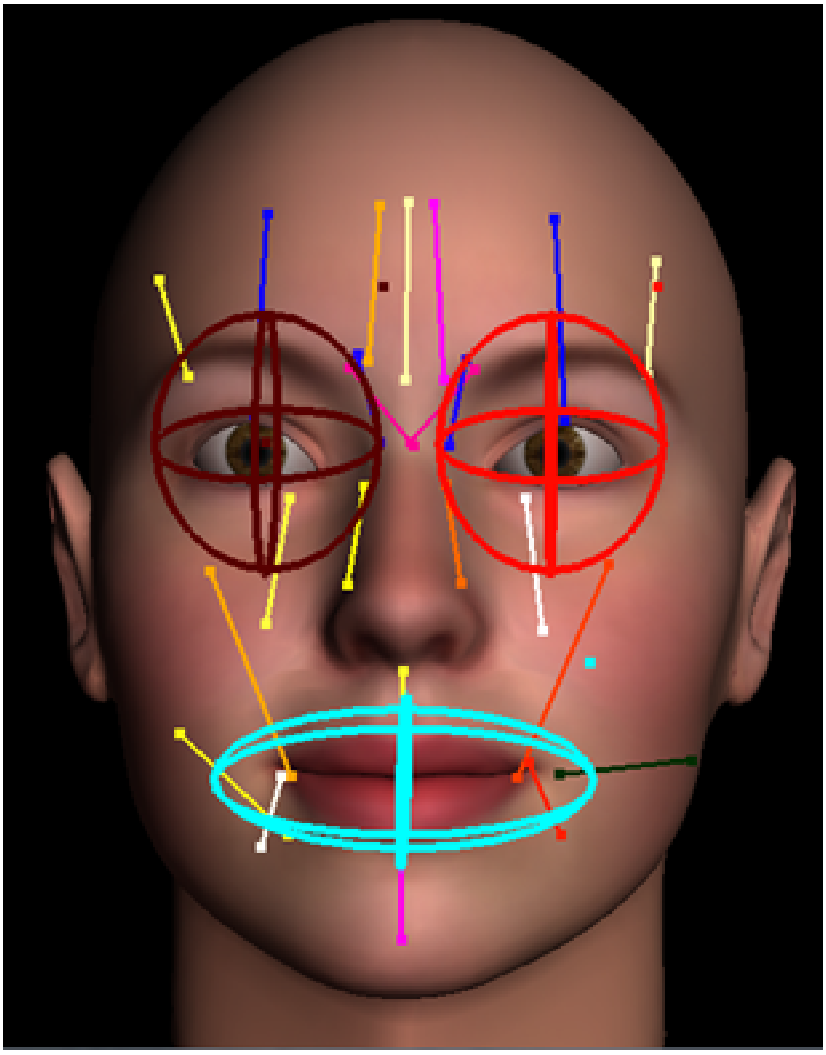


Figure 4.10: Visualization of pseudo-muscles usage; Muscles surrounding the nose, frontalis, and depressor muscles are the most deformed ones

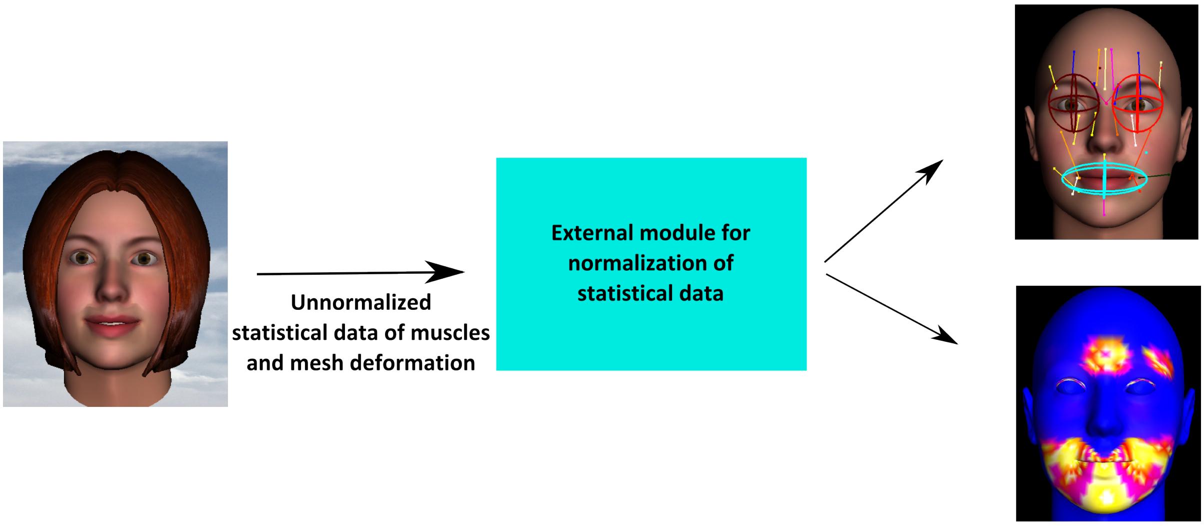


Figure 4.11: Implementation – data flow between modules

4.6.3 Implementation of Visualization Modes

Both visualization modes were incorporated into ECAF toolkit as internal and external modules. Firstly in standard mode the head receives commands from application and animates the head. During the animation we save statistical data for each vertex and muscle. The data output of this operation is unnormalized statistical data.

Then the statistical data are normalized by external modules. And finally the normalized data is the input for internal visualization module that displays the visualization on the talking head using heat-map color transformation. See the whole scheme of the implementation and data flows in the Figure 4.11.

4.6.4 Usage of Visualizations

Both implemented visualization methods have various applications and advantages in the process of exploring deformations of head mesh or muscles.

For example if one displays one head in standard size then there can be a visible stressed area. Subsequently displaying the same head as distant (very small); this area can be barely seen (see Figure 4.12). This feature possibly helps us in identifying much stressed areas that could be also simplified when the head is distant from observer. The same could be done in the area of muscles where frequently used but small muscles could be omitted with help of the visualization. The visualization is just a helpful tool to find out face parts that can be simplified.

Secondly, the visualization of muscle usage is useful for fast comparison of two different muscle deformation statistical sources (e.g. for timetable applications, news reading). Visualization is a faster method than comparison of the same data in tables.

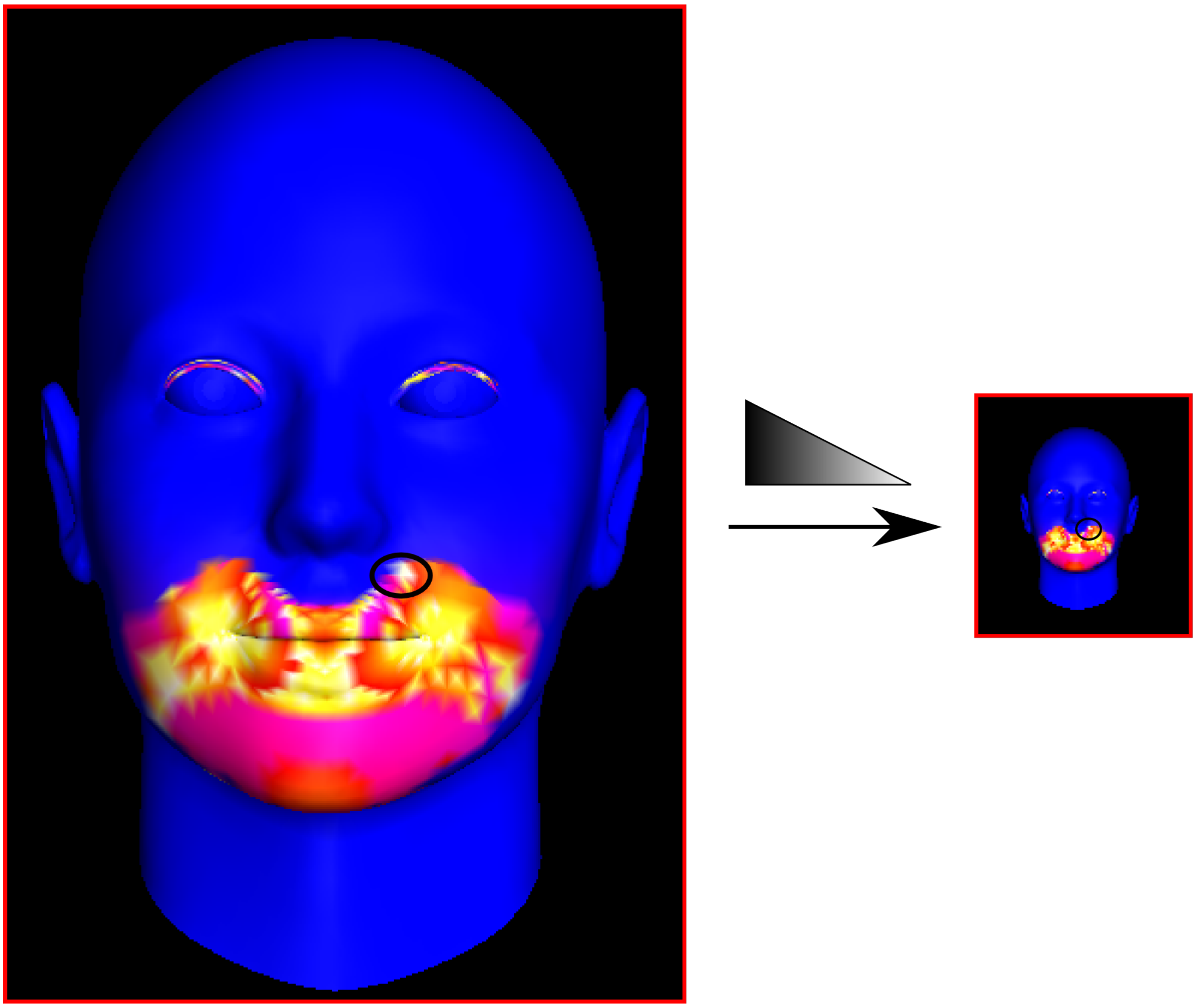


Figure 4.12: Using visualization to determine barely visible areas

4.7 Making the Dialog More Natural

This section describes the improvement naturalness of humanized user interfaces through several techniques. Human communication utilizes plenty of cues unconsciously or consciously. These cues display behavior or manage the communication channel.

The communication channel management cues can be used to make the artificial dialog more natural. Current speech dialog systems often need the human to press some push to talk button to take the turn and to put a speech dialog system to the listening mode. This can be improved by introducing turn-yielding cues in some cases where an ECA wants to yield a turn to the human. The procedure is analyzed in detail in the next paragraphs.

4.7.1 Introducing Turn-yielding Phenomena into Billionaire

The main challenge is how to introduce successfully turn-yielding phenomena into a spoken dialog system which uses concatenative speech synthesis to build a system which communicates with the user in a more natural way. One possible solution would be to introduce an ECA in such a dialog system. ECA is able to display/transmit the turn-taking/yielding cues without changing the voice. Having such an ECA-based dialog system would be useful for applications like a voice driven jukebox [36]. Comparing the potential of a virtual agent's visual turn-yielding cues to vocal ones is relatively little seen in research works. The model of communication is depicted in Figure 3.5 in Section 3.3. Taking into account the efficiency of communication the simple push-to-talk technology could be very efficient in a speech dialog system as Fernandez et. al. suggest [52]. But, on the other hand, it loses its interactive ability and can be associated with a "vending machine" syndrome. The syndrome can cause lower naturalness of interaction with a user [180].

Several turn-yielding cues are employed in the experiment. The selection consists of three vocal turn-yielding cues (pitch fall, higher speaking rate and loudness at the end of utterance) because of their effectiveness (see Table 4.2) and because they are also re-synthesizable. Two visual turn-yielding cues introduced are:

1. movement of head is slowed down before yielding the turn in dialog
2. small head nod at the end of utterance

They were selected because they can be seen on the face, i.e. the whole body is not needed for their expression. An eye-gaze based turn-yielding cue was not selected. As stated in the literature, eye-gaze based turn-yielding cues are very important in multi-party conversations [68] but they do not have as definite a role in two-party conversations [87].

4.7.1.1 Turn-Yielding Cues Parameters

This section lists the main parameters of how the turn-yielding cues are created. Table 4.3 shows vocal and visual turn-yielding cues creation parameters. Vocal cues are described by relative changes of frequency[3] F_0. Typically, they include relative speedup at the end

[3] Fundamental frequency is defined as the lowest frequency of a periodic waveform

Table 4.2: Table of implemented turn-yielding cues

Category	Turn-yielding cue
Utterance pitch (vocal)	Fall
Utterance final speed (vocal)	Higher
Utterance final loudness (vocal)	Low
Talking head movement	Stopped
Talking head nod	Head nod

Table 4.3: Audio parameters of turn-yielding cues

Cue	Pitch contour	F_0 change	Intensity change	Speed change
Pitch fall	%L-L	104-98Hz	0dB	The same
Final loudness	%H-L	182-88Hz	-10dB	The same
Final speed	%H-L	182-88Hz	0dB	1.6x faster
Visual cues	%H-L	210-108Hz	0dB	The same

of turn, pitch and a volume relative change in dB. The pitch contour pictogram follows ToBi annotation conventions [13]. Intensity compares sound intensity change at the turn-ending to overall previous intensity in dB. And finally, speed change represents a factor of changed speed at an utterance turn-ending. The length of all vocal turn-yielding cues was about 500 ms.

Visual cues are implemented using the talking head ECA mentioned in the previous section. The talking head movement cue is a very simple one. The talking head constantly moves according to a pseudo-random movement pattern. The cue is displayed as a stop of this movement just before turn-ending. It also has a length of 500 ms. The second cue, that of head nod, is represented by an animation 500 ms long where the head bends a little bit forward and then returns to the neutral position.

The turn-yielding cues seem to be one technique to improve the naturalness of ECA-based communication with human users. Further, the behavior patterns implementation is going to be introduced in the next section.

4.8 Extending ECAF Language – High-level Contextual Information

This section describes the extension of ECAF language by designing high-level commands that influence the behavior of the ECA in various contexts. The ECAF language allows to define fine-grained behavior by inserting gesture tags that can trigger the proper behavior of the ECA agent. However, the development of applications and mock-ups demonstrated that it is a tedious process to define the behavior on such a low level.

The developers of ECA applications need a database of behavior patterns that would be easy to use in various environmental, information, and mood contexts.

We propose a knowledge base that contains various patterns of ECA behavior for specific context of use. This database should extend the ECAF language so that it would still allow to define the behavior in a fine-grained manner as it is defined in previous sections.

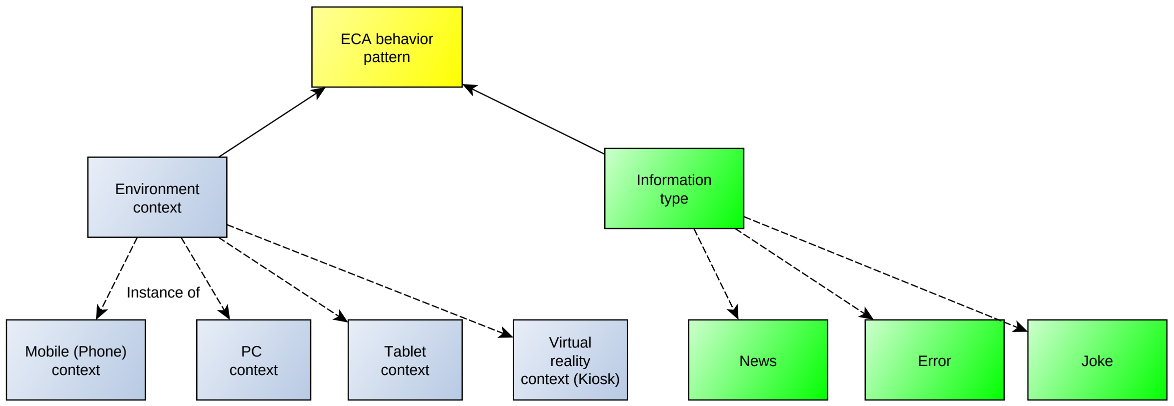


Figure 4.13: ECA behavior pattern ontological dependencies

The behavior can have multiple properties and dependencies (see Figure 4.13).

Various applications of talking virtual avatars need a number of proper high-level patterns. Some of these patterns are reusable in multiple categories based on current context. Binding the behavior to an overall context in which the behavior is presented has roots in human psychology. The theory of human action identification through exploring other people's behavior is described in [184]. Action identification is influenced by three factors: action context, action difficulty and action experience.

Without knowing the action context and simply knowing only physical movements, action can be hardly identified. E.g. solving the math equation can be seen as simple math practicing in the context of privacy. However, in the situation of school class, solving the equation can be categorized as showing the math skill.

The novelty of action can make one sensitive to other (lower level) features of such action, while mastering some subject can impose searching for higher level of action features (e.g. passes in soccer). For example, talking to a stranger can be more difficult than talking to a friend. Because the situation of talking to the stranger forces us to maintain the lower level behavior (e.g. sounding sincere, making continuous eye contact). The behavior pattern can be used to differentiate the talking head behavior among new users and users the talking head knows.

Action difficulty is often determined by the person's degree of experience with a specific action. There is a correlation between action experience and action automaticity [100]. People become familiar with lower level components of the action and these components are chunked into larger units. Further, these larger units continuously become units that are controlled by human mind (e.g. when driving a car, pedals controlling becomes automatic). Subsequently, for unexperienced people low level components of action are difficult to maintain. Thus, high-level components are practically blocked by human's mind.

Important part of the talking head behavior is the right insertion of pauses when presenting the text.

4.8.1 Conversational Cues Patterns

Human-to-human conversation is full of various cues that help the listener understand the speaker. Pauses are one of such cues. Pauses in a presented text are useful both for the speaker and for the listener [63]. Boundary pauses are the most frequent ones and they could be used to detect end of sentences if their use would be consistent. However, this is not true in all cases, non-boundary pauses are also part of spontaneous speech [201]. Pauses have multiple functions in human-to-human conversation. They are used for turn-taking, to emphasize some part of the speech and least but not last as a rhythmic guide.

The talking head can use the effect of dramatic pauses to emphasize a part of text that is important and needs to be conveyed to the user. The distribution of pauses can be defined differently for news reading (slower, dramatic pauses) and for handing over a simple message (faster, few pauses).

Another patterns can use different nature of conveyed message. When an ECA needs to tell a joke or something funny, it is convenient to express happy emotions, or smile at the time of a punch line. Moreover, important messages can be accompanied by colors in UI that are connected with the notion of importance (e.g. red).

This high-level extension of ECA authoring language can be also used to control the turn-yielding mechanism. When the dialog manager decides it needs to request something from the user (e.g. asking for confirmation), this is marked as a request prompt. Request prompt patterns can cause the turn-yielding cue to be used. The usage of turn-yielding cues can be contextually dependent on the surrounding environment (e.g. in the public spaces visual turn-yielding cues can be more convenient and audio turn-yielding cues can be used in private space). Noisy environments represent a challenge for spoken dialog applications (e.g. car environment or public spaces).

4.8.2 Patterns in Noisy Environment

A study of communication patterns in noisy environment was done in [A.8]. The participants of this study were connected by Skype-line and exchanged spoken messages with one another. Both were exposed to varying types and levels of noise. The output of this study is a list of interesting communication patterns humans use when exposed to noise. Some of these patterns can be useful for ECA humanized interfaces.

Confirmation can be done in various ways. Under certain circumstances simple acknowledgments (e.g. “OK”, “yes”, “got it”, “hmm”) are suitable. These are meant to be used by ECAs in relatively quiet environments. For environments that are noisier, acknowledgment by repeating the sentence could be more appropriate.

One general observation pattern comes from the study. The participants tended to use shorter segments of fluent speech in the noisy environment. This should be useful to replicate for ECA to convey messages to users in shorter way than in quiet environment.

4.9 Summary

This chapter dealt with solutions to the issues that exist in the world of humanized ECA speech-based interfaces. The realization of the proposal was incorporated to the ECAF Talking head toolkit. Especially, for the serious game “Who wants to be Billionaire”, which is targeted for older adults. The virtual persona of Anne and her key characteristics are introduced.

Firstly, the ECAF architecture is described including the ECA authoring language. The main advantage of the authoring language is that it combines means to control the low-level ECA behavior and "classical" GUI means (e.g. images, videos, buttons and touch-, click-handlers,...).

Further, the connection to the CIMA dialog manager is depicted. Some interesting framework properties are explained. These include mainly, prompting, disambiguation, action announcement and usage of context to define the behavior of speech dialog application (context as combination of user and system acts in one dialog package).

The design of the Billionaire game GUI is discussed and improvements based on the target group characteristics are depicted on the game screen shots.

The chapter touches even the deformation models used for the ECA rendering. Two methods that solve the ECA level of detail are realized, one is called Adaptive 3D Mesh Simplification, which is based on non-uniform deforming of the ECA. The second method Adaptive Reduction of Pseudo-muscles reduces the number of deformed muscles on the head adaptively according to the presented text and the resolution of displays. These methods are used on computers with not enough power to render the fully-defined ECA.

In order to make the dialog more natural turn-yielding cues are introduced to the ECAF toolkit. These cues allow to yield a turn to the user and s/he does not need to indicate this by keyboard or using other means. This is even more useful in the virtual-reality based environments like kiosks in public places.

Finally the chapter describes an extension of the ECAF authoring language by high-level means that allow developers to define the behavior of an ECA agent by telling the ECA the environment, the text context and user variables – then proper behavior patterns are selected and the application developer does not need to use low-level gesture tags of the ECAF language.

The next chapter analyzes the implemented features of the ECAF talking head toolkit using various evaluations that were done with human test participants.

5 Evaluation

This chapter describes several evaluations of improvements proposed by this thesis. The studies are mainly done by using human participants. The first study deals with the settings of ECA appearance parameters.

5.1 Evaluation of ECA Appearance Parameters settings

Some “dynamic” parameters seem to be more important to the user than other ones. Four possible behavior (appearance) parameters were chosen and investigated. The parameters were chosen to be modified in the ECAF toolkit system. A pilot user study is presented where selection of such parameters was evaluated by human participants. The goal of this experiment is to find out whether some behavior parameters can be more important than others. The set of inputs consisted of 15 ECA videos presented to participants. Using statistical analysis means we evaluated the human sensitivity to these parameters.

5.1.1 Experiment

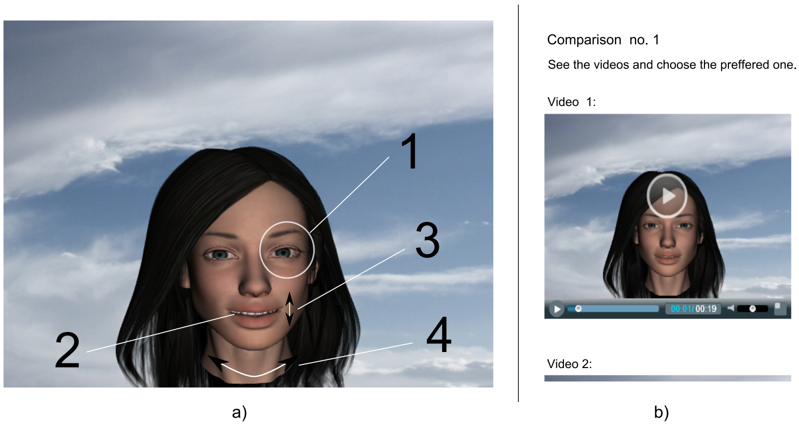
This section describes the details of the perceptual experiment which was conducted to evaluate human sensitivity to changing some parameters of ECA behavior. The psychophysical method of paired comparisons was exploited, two-alternatives forced choice [38]. During the evaluation four experiments were performed at the same time in order to evaluate four behavioral parameters. The experiment was a test of human preference, participants saw a pair of video files in one trial and chose the preferred one based on subjective evaluation. We would like to find out the user preferences (and sensitivity) to these four parameters. The metric of preference was subjective user judgment [124].

The four parameters that we want to evaluate and that will be modified during the generation of video files were selected. These parameters are summarized in Table 5.1. Figure 5.1 describes the parameters visually using a typical static image produced by the ECAF toolkit during speech.

First three parameters are intuitive, but the last one needs more explanation. The talking head in ECAF toolkit moves the head according to a simple up-down pattern. This animation is further enhanced by adding movement noise. The values of the last parameter represent only the speed of the final head’s movement animation.

Table 5.1: Four evaluated behavioral parameters of ECA (default value means ECAF toolkit default settings)

ECA Parameter	**Evaluated values**
eye’s blinking	no blinking, default ($p = 0.0085$), fast blinking ($p = 0.1$)
teeth color (static)	pure white (default), grey-yellow, darker grey-yellow
mouth opening (speech)	less (80%), default (100%), more (110%)
head movements (speech)	no movement, default, faster movements



Comparison no. 1

See the videos and choose the preffered one.

Video 1:

Video 2:

Figure 5.1: *a)* Evaluated behavioral parameters of ECA. Numbers: (1) eye's blinking, (2) teeth color, (3) mouth opening, (4) head movements. *b)* Web application used for carrying out the experiment – Comparison of two videos

5.1.2 Evaluation Videos

Fifteen videos generated by ECAF toolkit were used for the test. Videos were relatively short (19 seconds) in order not to exhaust human judges. The talking head was looking at the listener all the time, as it is visible in Fig. 5.1 a. In every video the talking head said the same sentence not to bias the results by choice of the sentence. The sentence was part of a news message: *"White House officials and some members of Congress reacted strongly Sunday to news that insurance giant AIG had intended to pay out 165 million dollar in bonuses and compensation. The company has received at least 170 billion dollar in federal bailout money."*. The only one difference in these videos were parameter settings (see Table 5.1).

5.1.3 Experimental Setup

The experiment was conducted remotely. This allowed to watch the videos in the quiet and private environment of home. Every participant compared the videos on his own computer in different environment. Every video had resolution 640 × 480 and participants saw the videos in this resolution (it was not possible to scale down or up). In total 93 participants took part in the comparison experiment. The participants were both male and female in between ages of 20 to 24, university students of computer science. Before the experiment every participant was instructed how to proceed with comparison. The whole experiment was anonymous. The next section will properly describe the experimental routine.

5.1.4 Experimental Routine

As it was said in Section 5.1.1, a method of paired comparisons was followed, especially two-alternatives forced choice. A simple web application was developed. Before using this web application the participants were verbally instructed how to proceed with the experiment. The first page of web application contained a login field and written instructions. After login the participant was navigated to the comparison test. There are

Table 5.2: ANOVA test of data sets (F-value critical for data sets $F = 3.028$).

Parameter	F**-value**	p**-value**
eye's blinking	5.240	0.006
teeth color	3.988	0.020
mouth opening during speech	15.289	10^{-7}
head movements during speech	35.965	10^{-14}

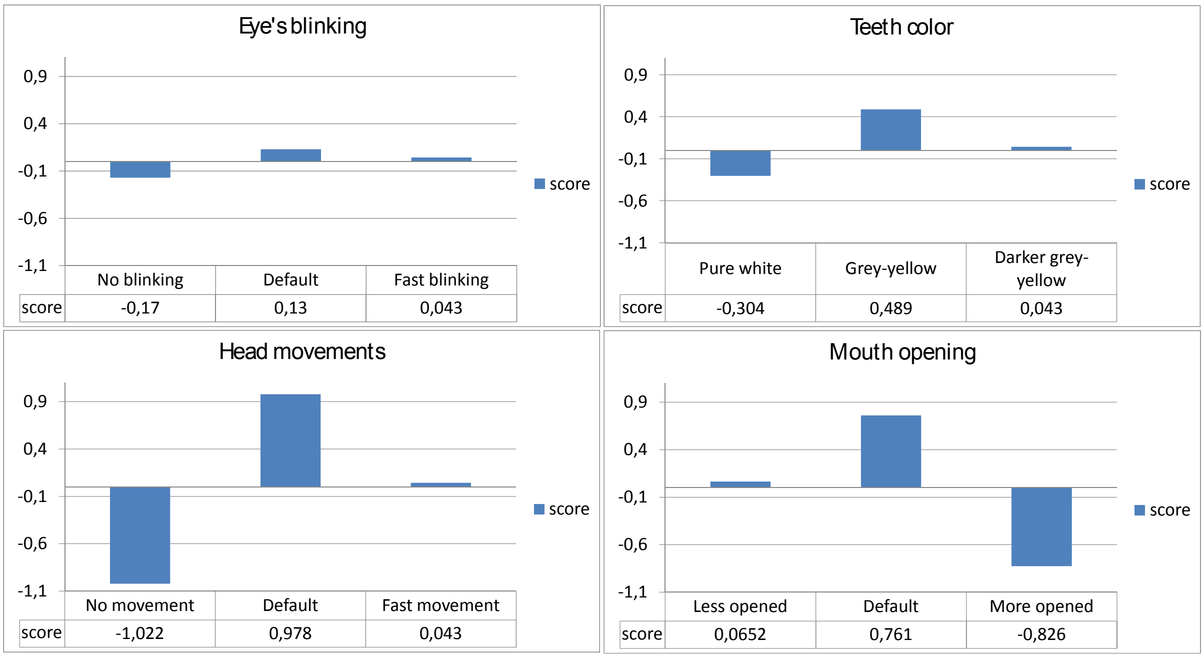


Figure 5.2: Graphs of calculated standard score for all four behavior parameters (positive values – more preferred)

4 parameters (see Table 5.1) and every parameter has 3 values. Each value of parameter was compared with each other value of the same parameter (not mixing them). We have $n(n-1)/2 = 3(3-1)/2 = 3$ comparisons for each parameter ($n = 3$ as three values of parameter). In total we prepared 12+3 comparisons. The +3 are three test comparisons that have interposed position of videos to address order of playing. The order of 15 trials was randomized in two selected test sets (A and B). This should ensure that the order of comparisons (trials) does not influence results.

Every comparison of two videos was depicted on a web page in integrated Adobe Flash video players. Participants were asked to play both videos in the order, see Fig. 5.1 b. After they saw the video they should answer a simple question: "Which video do you prefer more the first or the second one?" and select the appropriate answer by a radio button.

Each participant totally completed 15 comparisons and he/she was not time limited. The experiment could be interrupted anytime and continued later. This should not affect the results as all pairs of videos are independent.

5.1.5 Results and Discussion

In total 93 participants completed the experiment. They carried out 1380 pair comparisons. One part (47.3%) of participants took the comparison set A and second part (52.7%) took the set B. For each pair comparison the video chosen by participant was given score 1 and the second video (not chosen one) was given score 0. These data were filled in 3×3 matrix for each parameter. So from one participant we have four 3×3 matrices of scores. The matrix columns and rows represent values of a particular parameter. The Thurstone's Law of Comparative Judgment – Case V [182] will be used to convert the (0-1) scores into standard scores that are comparable. The graph comparison of standard scores for each parameter is visible in Fig. 5.2.

The standard scores are normally distributed. This warrants us to use classical statistic methods to further analyze the results. We used ANOVA (Analysis of variance test) to analyze statistical significance of collected data. See results in Tab. 5.2. The test shows that all four experiments gave out statistically significant results ($F > F_{critical}$ and $p < 0.05$).

As Figure 5.2 shows in graphs, participants seem to be less sensitive to eye's blinking frequency than to the other parameters (the scores are very similar). But the other three parameters (teeth color, head movements and mouth opening) indicate some form of human sensitivity to them, the preference scores of parameter values are different. The teeth color parameter shows more similar score in comparison to both of the movement parameters, changing their values is most visible. These parameters seem to be good starting point in personalization of dynamic appearance parameters. Dynamic appearance in the dialog context seems to be important too. The next section describes evaluation of visual turn-yielding cues using an ECA.

5.2 Evaluation of Visual Dialog Turn-taking/yielding for ECA

The evaluation study is focused on testing two hypotheses:

Hypothesis H1 Using more turn-yielding cues before a transition relevance place increases the probability of the correct judgment about the next speaker. The turn-yielding cues can be both vocal and visual.

Hypothesis H2 Visual turn-yielding cues are better than vocal cues in increasing the probability of a correct judgment of who will be the next speaker.

5.2.1 Method

To test the delineated hypotheses mentioned above a perceptual experiment is introduced. An investigation of simple and complex turn-yielding signals is done in this experiment where participants watch and listen to videos of conversation between a talking head avatar as one of the dialog partners (Annie) and a simple vocal dialog partner (David). The talking head is used in order to find out whether this sort of avatar is capable of turn-yielding signaling and whether people can judge this signaling correctly.

The conversations are paused in selected spots and participants try to decide whether the speaker changes or not. The method of analysis of judgments of non-participating dialog listeners is exploited. The methodology of our perceptual experiment is mainly inspired by the previous work and experiments of Hjalmarsson because of the similarity of investigated issues [76]. Her work was later extended to experiments with synthesized voice. Hjalmarsson found out that synthetic voice affects judgments in a similar way as the human voice [77].

5.2.2 Experiments

Details of the perceptual experiment conducted with the aim to evaluate three vocal and two visual turn-yielding cues are provided in Table 2. The experiment was performed in two parts: first, the main experiment, and then a post-test experiment.

5.2.2.1 Dialog Data

The dialog part of the experiment was conducted in English because the speech synthesizer (IBM ViaVoice speech synthesizer) used supports English only both in formant and concatenative synthesis. Dialog data is needed to run the experiment successfully. As the speech dialog systems research is a relatively young research area there are no standard dialog data which could be used.

There are several possible ways to collect dialog data. First, one can design an experiment (a game) in which participants are forced to talk to each other while playing. Such a situation creates an environment where a natural dialog can be captured. Second, movie scripts serve as another source of dialog data from artificially prepared dialogs by skilled writers.

A part of a theater play script is chosen in this experiment to study turn-yielding cues, using artificially prepared dialogs from a play “The Importance of Being Earnest” by British author Oscar Wilde. The dialog data source for the second post-test experiment tries to counterbalance this and uses natural dialogs source. Dialog data for the post-test experiment are based on the Map Task Corpus of the University of Edinburgh, the part accessible from Dialogue Diversity Corpus[1]. These dialogs are not artificially written. They are collected during a map navigation game [2].

5.2.3 Experimental Design

The dialog data should be transformed from text to speech in order to evaluate turn-yielding cues. The speech part of this transformation is done through a speech synthesizer. The synthesizer is able to generate all turn-yielding vocal cues but the pitch fall cue; that only through formant synthesis. Wherever the pitch fall cue is used, the formant synthesis is present. The other dialog sequences use concatenative synthesis. The usage of formant synthesis which is not as good as concatenative synthesis is a possible problematic point. It is further discussed in Section 5.2.12. Visual cues are displayed by a talking head ECA application.

5.2.4 Evaluation Videos

Fifteen dialog scenarios were prepared and generated by the ECAF toolkit. Table 5.3 contains the list of video, audio and dialog parameters. The directions of dialog turns could not be counterbalanced correctly because David is just a voice and does not have

[1] `http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.htm`

Table 5.3: List of evaluated video/dialog parameters in the main experiment

Dialog no.	Turn-yielding cues	Direction of dialog turn-yield	Sound only
0	without	David – David	No
1	without	David – Annie	No
2	without	David – David	Yes
3	head movement	Annie – David	No
4	pitch fall	David – Annie	Yes
5	final speed	David – Annie	No
6	final loudness	David – Annie	Yes
7	head nod	Annie – David	No
8	final loudness, pitch	David – Annie	Yes
9	head movement, nod	Annie – David	No
10	final speed, pitch	Annie – David	Yes
11	head movement, nod, final speed	Annie – David	No
12	head movement, nod, final pitch	Annie – David	No
13	head movement, nod, final loudness, speed, pitch	Annie – David	No
14	head movement, nod, final loudness, speed	Annie – David	No

an avatar. He could yield turn just in sound only sequences. However this should not influence the results. Kennedy and Camden did not found significant gender differences in similar area of interruptions during conversations [95]. Each dialog sequence runs about 20 to 60 seconds. The sentences in dialog sequences were semantically and syntactically complete. The list of utterances just before the judgment points is in Appendix A.

5.2.5 Dialog and Experiment Parameters

Dialog partners are called “Annie” and “David”. Therefore, the dialog represents a man-to-woman conversation. Annie’s part is played by the talking head avatar and David is invisible (only his speech can be heard). During the preparation of dialog sequences, man and woman speech synthesizer voices were used to differentiate the roles of Annie and David. The dialogs were prepared to be interrupted during or at the end of Annie’s or David’s utterance. In this TRP point either Annie or David holds the turn or yields it to the partner.

Figure 5.3 depicts arrangement of the experiment. The test was driven by Adobe Flash web application (for screen shots see Figure 5.4) which played each dialog video to the participant and asked questions. The precise procedure of the experiment is given in the next section. Each of the videos is 640 x 480 pixels (190 x 142.5 mm) large and participants saw the videos in this size (it was not possible to scale down or up).

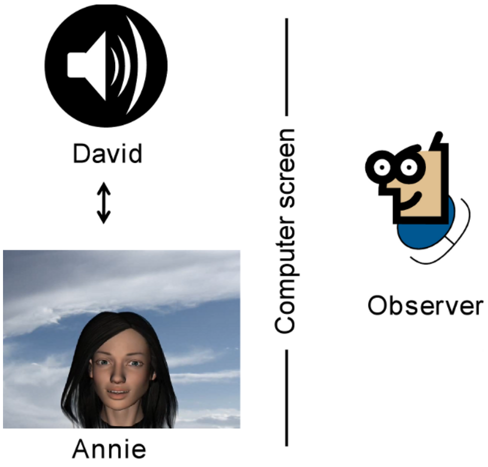
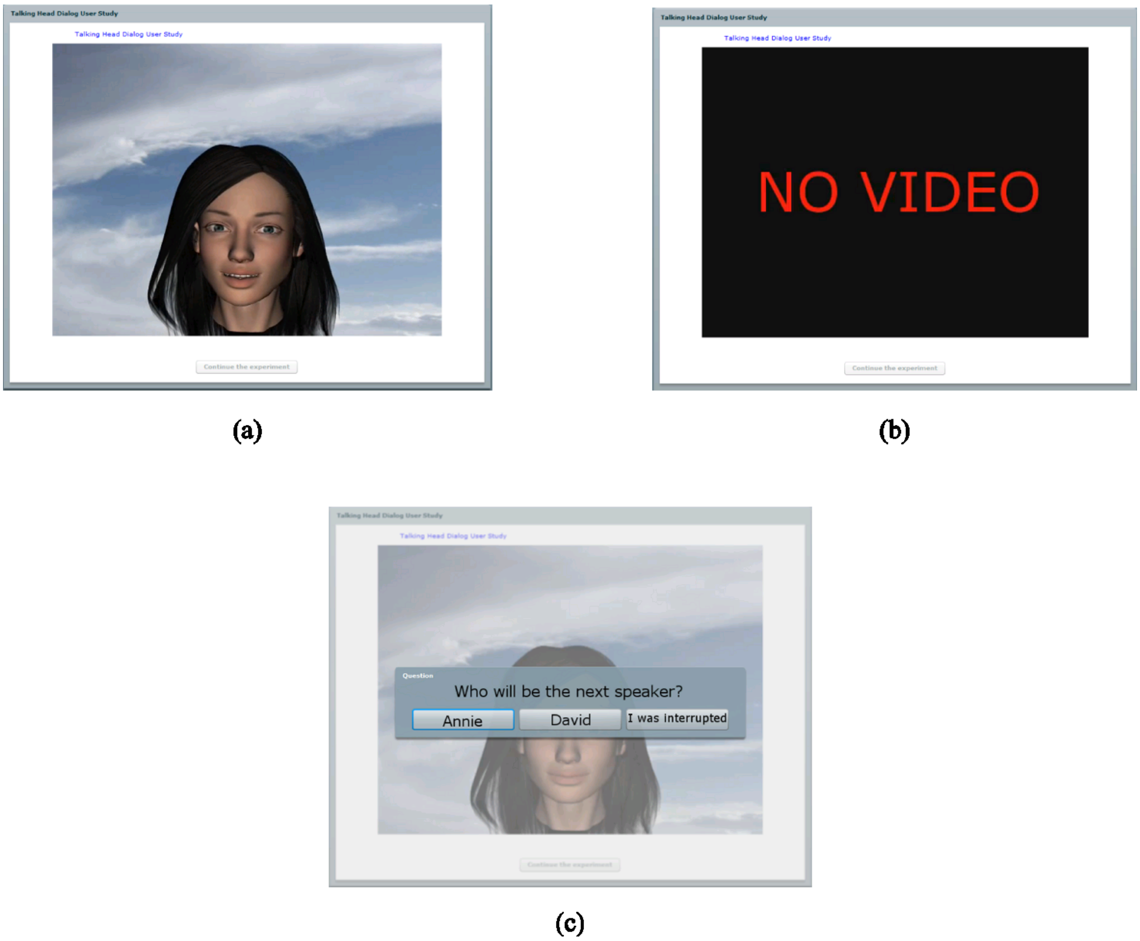


Figure 5.3: Experiment set up. Annie is the ECA with a synthesized voice and she talks to David (he is just synthesized voice). The observer follows a dialog sequence between Annie and David and this dialog sequence is interrupted. The observer judges who the next speaker will be (Annie or David) after the dialog sequence is interrupted.


Talking Head Dialog User Study
Talking Head Dialog User Study
Continue the experiment
(a)
Talking Head Dialog User Study
Talking Head Dialog User Study
NO VIDEO
Continue the experiment
(b)
Talking Head Dialog User Study
Talking Head Dialog User Study
Question
Annie
David
I was Interrupted
Continue the experiment
(c)

Who will be the next speaker?

Figure 5.4: Experiment web application screen shots: a) Talking head dialog video, b) Sound-only dialog sequence example, c) Question: “Who will be the next speaker”?

5.2.6 Experiment Procedure

The experiment was conducted remotely. Every participant observed the dialog sequences on their computer in a different but quiet environment at home in their spare time. How to proceed the experiment instructions were given verbally in a teaching lab. The possibility that one person would perform the experiment multiple times was excluded by the usage of faculty login name.

The method of non-participating judgments was followed. A Flash web application was developed, mainly because of the ability of video playback. The first page of the web application contained the login field (to exclude multiple participation) and written instructions. After login, the participant navigated to the first video dialog. The participant should see the screen with video (Figure 5.4a) and hear the sound, in case the sequence was recorded with a talking head. In the case of sound-only dialog, depicted in Figure 5.4b, the participant heard only the sound of the dialog. The "NO VIDEO" (Figure 5.4b) screen was shown to the participants before the experiment. This sign should not affect the results because it is known in advance. Finally, the particular turn-yielding cues are presented (or not) and the video suddenly ends. The participant is asked the question: "Who will be the next speaker? Annie or David?" (see Figure 5.4c). The possible answers are: "Annie", "David", or "I was interrupted". The last option allows the participant, in case he/she was interrupted during the video observation (for example, by a phone call, another person, etc.), to repeat this particular dialog case. After he/she answers the question he/she can proceed to the next dialog sequence. The "I was interrupted" allows a participant to repeat the video. This could invalidate the whole experiment for that participant, but nobody used this "I was interrupted" button.

Each participant judged 15 spots in dialogs (one judgment spot at the end of each dialog sequence). During the test, it was possible to suspend the procedure between dialog sequences and finish it later. Random order of dialog sequences was generated for each participant to counterbalance the learning curve effect. Correct answers were not revealed to participants during the experiment procedure. They were revealed to them after all participants successfully conducted the experiment.

5.2.7 Experiment Evaluation and Discussion

In total 40 participants completed our experiment. Participants answered a total of 600 questions (mentioned above). The application recorded one answer for each question. The possible answers for each question are: "Annie" or "David". The results were converted to the graph in Figure 5.5. The horizontal axis shows all videos and the vertical axis represents the percentage of correct answers.

The results were analyzed for gender differences. Chi-square test with Yates continuity correction was applied to each dialog sequence. The tests revealed that there are no significant differences between the genders (all $p-$values were > 0.1). The largest difference was in the case of dialog sequence 5 (final speed, $p = 0.11$), nevertheless, this difference is still not significant.

It can be seen that only in dialog sequences 0, 1, 4 and 8 the majority of participants judged incorrectly. Sequences on the right side of the graph show very good results. According to Table 5.3, these dialog sequences used mostly combinations of cues. To evaluate both hypotheses the experimental data were statistically analyzed.

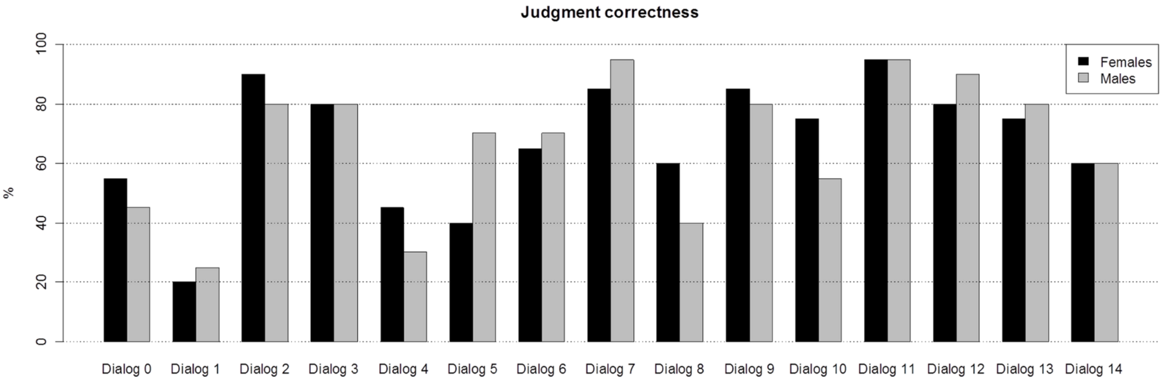


Figure 5.5: Judgment correctness of participants in the main experiment. The results of male and female participants were expressed separately to reflect possible differences between genders. The most correct answers were from dialog sequence no. 11. A combination of three turn-yielding cues (head movement, head nod and final speed) was used there. See Table 5.3 for other dialog sequences description

Table 5.4: Odds ratios of how correct judgment improves with more cues used

Number of cues	Odds ratios	p-value
1	0.51	$p = 0.06$
$\langle 2, 4)$	0.98	$p = 0.00$
$\langle 4, 5\rangle$	0.64	$p = 0.05$

5.2.8 Hypothesis H1

As said at the beginning of the Experiment Section, the hypothesis H1 examines the usage of turn-yielding cue combinations in one TRP. The decision process after each of the dialog sequences is in fact a Bernoulli trial because of the two options as to who will be the next speaker [138]. Having the number of used turn-yield cues in each dialog sequence, logistic regression statistics could be used. We used a binary ordinal logistic regression model and all statistical computations were done using the R software [152].

The results of logistic regression show that there is a relation between the number of used turn-yielding cues and correctness of participant judgment. Table 5.4 shows coefficients on how the correctness of judgment changes when a particular number of turn-yielding cues is used (odds ratios). The counts of turn-yielding cues are split to three groups to lower degrees of freedom and to provide better fit of a final model. Three results are statistically significant and they improve the correctness of judgment substantially. It supports the hypothesis H1.

Likelihood ratio test χ^2 of 12.84 with 3 degrees of freedom and $p = 0.00$ show that our model fits significantly better than empty (null) model. We dont report Cox & Snell R^2, Nagelkerke R^2 or other R^2 measures because they do not assess goodness-of-fit [81]. R^2 measures are based on comparisons of predicted values from the fitted model to intercept or null model only; however they may be helpful in the model building state for evaluating competing models [81]. Measures of fit are based on a comparison of observed to predicted values from the fitted model. The correct classification table is superseded by the following goodness-of-fit test.

5.2.9 Goodness-of-Fit Test

The Hosmer-Le Cessie statistic test is a measure of lack of logistic regression model fit. This statistic test is based on the Hosmer-Lemeshow test, which divides the data into groups of equal size, and then compares the observed to expected number of positive responses and performs a χ^2 test. The Hosmer-Le Cessie test solves some insufficiencies of the original test [80]. The test was computed for the hypothesis H1 data. Unweighted sum-of-squares is 118.95, expected mean value is 118.95, variance < 0.00001 and $p = 1.00$. Very large value of sum-of-squares would indicate lack of fit, but this value seems to be low enough, so the logistic regression model does not show a lack of fit. The p-value is high enough not to reject the null hypothesis that the data follow the logistic regression model (in Table 5.4).

5.2.10 Hypothesis H2

The hypothesis H2 says that visual cues are better than vocal ones. Logistic regression can also validate this hypothesis when the experiment results are separated to individual variables. Figure 5.6 shows the results of logistic regression and 90% confidence intervals. The pitch fall cue ($p = 0.24$), the head movement cue ($p = 0.15$) and the final speed ($p = 0.61$) results are not significant.

Figure 5.6 can be viewed that the hypothesis H2 is valid. The potential of visual cues appears to be better. However, the results of this experiment are misleading because the dialog sequences contain a mixture of visual and vocal cues. Better insight can be provided by another experiment which will have visual and vocal cues separated. Likelihood ratio test χ^2 of 45.38 with 5 degrees of freedom and $p = 0.00$ shows that our model fits significantly better than empty (null) model. The Hosmer-Le Cessie goodness of fit test turned out to be as following: Unweighted sum-of-squares is 112.49, expected mean value is 112.23, variance 0.21 and $p = 0.19$. The p-value is reasonably high and the model fits well. The experiment in the next section can bring clearer view on hypothesis H2.

5.2.11 Post-test Experiment

The post-test experiment focuses on hypothesis H2 and brings a less distorted view on this hypothesis in that the visual and vocal cues are not mixed in dialog sequences. The conditions and execution of the experiment are exactly the same as during the experiment in Section Experiment. Although the number of participants is smaller than in the previous experiment, it is sufficient for us to show validity of the hypothesis H2 as the findings are statistically significant. The main difference is the usage of the Map Task dialog source mentioned in Section Dialog Data which brings real dialog utterances in contrast to Oscar Wilde's play.

The video/audio dialogs were created in the same way as Annie's and David's synthesized speech. The parameters and dialog setup could be found in Table 5.5. Ten dialog sequences were prepared and used the same dialog in all sequences. This time when the

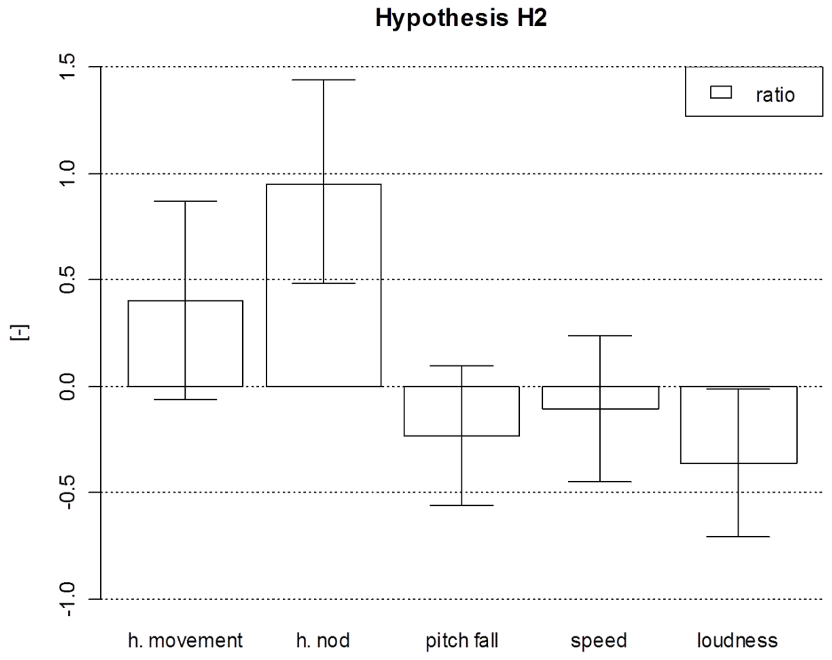


Figure 5.6: Graph of the logistic regression results for hypothesis H2. It shows ratios how probability of correct judgment rises or falls using particular turn-yielding cue. Visual turn-yielding cues (head movement and head nod) raise the probability. Vice versa the vocal turn-yielding cues decrease or leave the same probability of correct judgment.

vocal cues are utilized in a dialog, only the audio is evaluated. Naturally, visual cues required video dialogs. Concatenative synthesis was used for all dialog sequences and the pitch fall cue was simulated using the Praat tool (http://www.fon.hum.uva.nl/praat/).

As in the previous experiment, sentences in dialog sequences were semantically and syntactically complete. List of the utterances preceding the judgment point are in Appendix A.

This time the experiment was conducted with a total of 31 participants. The participants were males (65%) and females (35%) mean age 26.60 (SD 3.60). Not a single participant took part in the previous experiment. This group of participants was more heterogeneous than the previous one. There were not only students involved. Participants had Czech cultural background. We wanted to compensate for the previous homogeneous group of students and examine the results in a different group of participants. The participants went through the prepared dialog sequences.

The results of judgment correctness are summarized by the graph in Figure 5.7. The results of the turn-yielding cues of head movement, head nod and sound loudness are statistically significant. The results of final speed and falling pitch are not statistically significant. See Table 5.6 for particular odds ratios. As it can be seen, the odds ratio of vocal cue loudness improved in comparison to the previous experiment, but the final pitch still has a negative effect on judgment correctness (in this experiment not statistically significant). The improvement due to the usage of loudness vocal cue could be realized by different dialog sequences and by the usage of qualitatively better concatenative synthesis

Table 5.5: List of post-test evaluation video/dialog parameters

Dialog no.	Turn-yielding cues	Direction of dialog turn-yield	Sound only
0	without	David – David	No
1	without	Annie – Annie	Yes
2	head movement	Annie – David	No
3	final speed	Annie – David	Yes
4	head nod	Annie – David	No
5	final loudness	Annie – David	Yes
6	head movement, nod	Annie – David	No
7	pitch fall	Annie – David	Yes
8	without	Annie – Annie	No
9	final loudness, speed, pitch fall	Annie – David	Yes

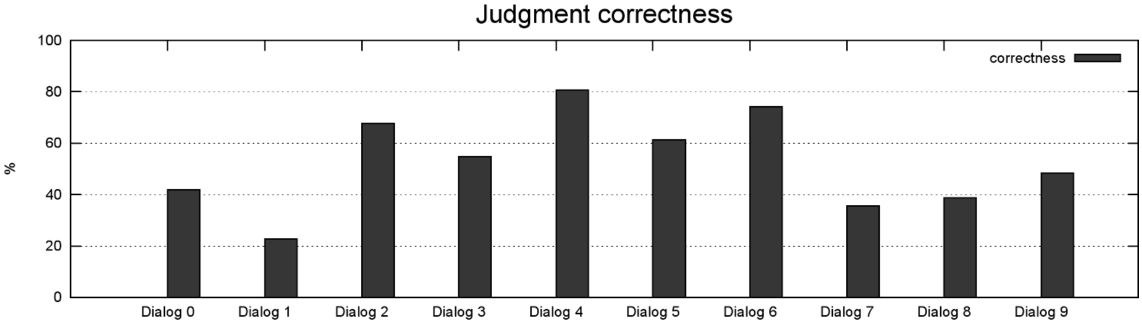


Figure 5.7: Judgment correctness of participants in the post-test experiment. See Table 5.5 for dialog sequences description.

in comparison to the formant synthesis. Likelihood ratio test χ^2 of 29.72 with 5 degrees of freedom and $p = 0.00$ shows that our model fits significantly better than the empty (null) model. The Hosmer-Le Cessie goodness of fit test was also done and the values are the following: unweighted sum-of-squares is 71.51, expected mean value is 71.87, variance 0.15 and $p = 0.06$. There does not seem to be a lack of fit. The statistical non-significance of two vocal turn-yielding cues seems to also be caused by very indecisive results for these cues. But the overall judgment correctness of our selection of visual turn-yielding cues supports the hypothesis H2 that they are more reliable than the vocal ones.

5.2.12 Discussion

Two non-participating judgment experiments were performed and five visual and vocal turn-yielding cues were compared. A total of 71 participants accomplished both experiments. Each participant judged 15 dialog sequences in the first experiment or 10 dialog sequences in the second one. Findings from the first main experiment suggest that the hypothesis H1 (Using more turn-yielding cues before a transition relevance place increases the probability of correct judgment of the next speaker) is valid. Possible gender differences in the results were also analyzed. The results of statistical tests show that

Table 5.6: Odds ratios of how correct judgment improves with more cues used. Statistically significant results are in bold

Turn-yielding cue	**Odds ratios**	**p-value**
Head movement	0.71	$p = 0.04$
Head nod	1.28	$p = 0.00$
Pitch fall	-0.46	$p = 0.17$
Final speed	0.36	$p = 0.28$
Loudness	0.61	$p = 0.05$

there are no significant differences in turn-yielding cues perception between male and female participants.

The second hypothesis (H2) that visual cues are more reliable than vocal turn-yielding cues was also validated by the first experiment. However the pitch fall cue, the head movement cue and the final speed cue results are not significant. That could be caused by more indecisive judgment results. Not to be misled by the mixture of visual and vocal turn-yielding cues, the post-test experiment was prepared. The post-test experiment validated the second hypothesis. It turned out that our selected visual turn-yielding cues were more reliable than the vocal ones (H2) in the post-test experiment. Finally, the two experiments suggest that usage of selected visual turn-yielding cues has a positive impact on dialog turn management (better turn-ending estimates) in the area of two-party dialogs.

The findings concerning vocal turn-yielding cues are a little bit surprising. The selected vocal turn-yielding cues seem to have little impact on correct turn-ending judgment. Furthermore, pitch fall cue had a negative effect on this judgment in the first experiment as well as in the second experiment, however, the findings were not statistically significant. These results are in contradiction to previous studies considering vocal turn-yielding cues (e.g. [67]; [76]; [77]). One possible explanation could be the length of audio turn-yielding stimuli. According to previous studies of Barkhuysen et al. audio-only features are better classified when they are longer [11]. Or, in the case of the first main experiment, this could be caused by a formant synthesis quality.

Comparison of vocal and visual cues may seem a little bit “unfair” because the voice in the experiment had several functions (e.g. turn-taking and delivering intelligible speech). But the talking head had also several functions like the voice, turn-taking and delivering intelligible speech in form of face animation.

Although the post-test experiment tried to solve some shortcomings/weaknesses of the first experiment (e.g. vocal and visual cues mixed in dialogs, formant speech synthesis, etc.), some confounding points could still exist. Those should be addressed in future work. For example, the experiments were not fully gender or dialog modality counterbalanced because of non-existent male avatar. We used human-like female avatar only. Therefore, the results cannot be generalized and are limited to the conditions of the two experiments.

Still, the results of the experiments show clearly the advantage of ECA usage in speech dialog systems. The spoken dialog system architects have the ability to include turn-yielding cues of ECA and use state-of-the-art concatenative synthesis together in their systems. This could make the system more natural and improve its interactivity. Efficiency of interaction seems to be very good even in push-to-talk systems [52].

Table 5.7: Stimuli audio-video combinations prepared for participants. TTS stands for text-to-speech audio

Visual	Sound ba	Sound da	Sound ga	TTS ba	TTS da	TTS ga
Human	Visual ba	Visual ba	Visual ba			
	Visual da	Visual da	Visual da			
	Visual ga	Visual ga	Visual ga			
Talking head	Visual ba	Visual ba	Visual ba	Visual ba	Visual ba	Visual ba
	Visual da	Visual da	Visual da	Visual da	Visual da	Visual da
	Visual ga	Visual ga	Visual ga	Visual ga	Visual ga	Visual ga
Black screen	N/A	N/A	N/A	N/A	N/A	N/A

5.3 Evaluation of McGurk effect in ECAF Talking Head Toolkit

This section describes the procedure that was followed when analyzing influence of corrected vision to the McGurk test. The test is used as one of the pronunciation tests for virtual agents.

5.3.1 Experiment

This section describes details of perceptual experiment conducted with aim to evaluate how corrected-to-normal vision influences perception of the McGurk effect for a talking head. The experiment should validate or reject our hypothesis:

People with corrected vision will judge the McGurk effect differently than people with non-corrected normal vision.

Participants saw stimuli videos with synthetic McGurk effect and control sequences without McGurk effect. Totally 33 stimuli videos were prepared for participants. Videos contained possible combinations of artificial face, human face, text-to-speech audio and human audio modeling syllables *ba – da – ga* (see Table 5.7).

The video sequences in human category were recorded by the means of usual web camera in resolution 640 × 480. Figure 5.8 illustrates what portion of face was recorded. Audio tracks were recorded using computer microphone headset. Recorded visual video sequences were mixed with particular audio tracks in video editing software.

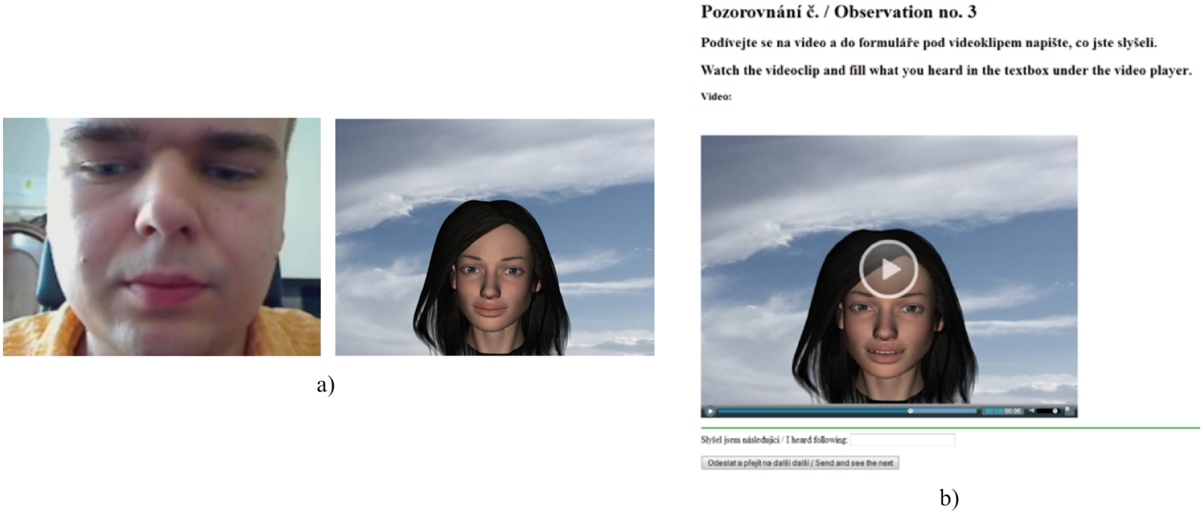
The synchronization of audio track and phonemes animation of McGurk sequences was modified manually.

5.3.2 Experimental Procedure

We prepared 33 video sequences (see Table 5.7) generated either by the ECAF toolkit or as recorded human sessions. Each sequence is about 7 seconds long.

The talking head or a human repeats 6 times a particular syllable ba, da or ga. The experiment was conducted remotely. Every participant observed the video sequences in his/her own computer in different but quiet environment.

The test was handled by a PHP web application with embedded Adobe Flash player for video playback (see Figure 5.8). It was not possible for participant to scale down the video from original resolution (640 × 480).



Pozorovnání č. / Observation no. 3

Podívejte se na video a do formuláře pod videoklipem napište, co jste slyšeli.

Watch the videoclip and fill what you heard in the textbox under the video player.

Video:

Slyšel jsem následující / I heard following:

Odeslat a přejít na další další / Send and see the next

Figure 5.8: Experimental web application a) example of human and talking head video sequence snapshots; b) the screen shot of web page with observation no. 3 which was seen by some participant. There is a text box for participant's answer under the video player window.

The participants were both males and females between ages of 23 to 26, university students of computer science. The whole experiment was anonymous and participants were instructed how to proceed with the experiment verbally. At the beginning the web application asked participants whether they have or not have some vision correction. After this question the participant was navigated to the first video sequence. For each participant random order of video sequences was generated. The first three video sequences were training ones and the data from them was not used. Participant observed each video and filled-in a text box with the text that he/she perceived. Each participant observed 36 (3 training ones + 33) video sequences.

5.3.3 Experimental Evaluation

In total 32 participants took part in our experiment (41% had corrected vision). The data from the experiment were manually normalized. For example: One participant answered vaaa vaa va and the second participant answered bababa. The correct answer based on audio track should be ba-ba ba-ba ba-ba. After normalization the first answer was marked as correct and the second as incorrect.

The graph in Figure 5.9 shows results for the McGurk sequences with the highest number of incorrect answers. Confusion video sequences are our main interest. The mean of incorrect responses for participants with normal vision is 3.89 (standard deviation 3.2). The mean of incorrect responses for participants with corrected-to-normal vision is 1.56 (standard deviation 1.85). This result denotes some differences in the McGurk effect perception by normal vision participants and participant with corrected vision. To make the results complete, sequences with no confusion produce mean of incorrect responses 1.2 (standard deviation 1.1).

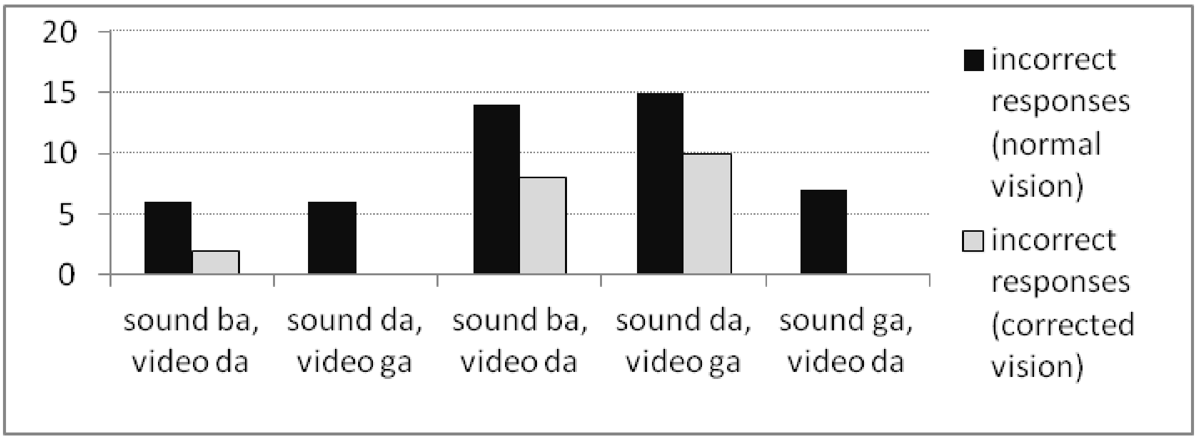


Figure 5.9: Results of talking head McGurk sequences with the highest number of incorrect answers. Last three results used text-to-speech generated audio.

5.3.4 Discussion

An experiment with the McGurk confusion audio-video sequences was conducted using talking head and human head. The hypothesis was that people with corrected vision will judge the McGurk effect differently than people with non-corrected normal vision. The previous section presented results of the experiment and it is clearly visible that there are some differences between the group of participants that had corrected-to-normal vision and participants with non-corrected normal vision.

Results of our work could help researchers that want to evaluate their developed talking heads using the McGurk perception test. They should consider vision correction means of their participants in their results and report both groups separately to be comparable.

5.4 Talking Head – Level of Detail Evaluation

In the previous chapter two methods of level of detail were introduced. The methods were implemented in the ECAF Talking Head toolkit. The methods are called Adaptive 3D Mesh Simplification and Adaptive Reduction of Pseudo-muscles.

In this section five experiments are introduced. The experiments show the usage of visualization and statistical classification of the muscles. The experiments are both generic and specialized. The specialized experiments explore the deformations and usage of muscles and head model in the areas of specific applications as weather forecast or application for timetable announcement. Firstly, a basic experiment is described.

5.4.1 Basic Experiment

This experiment assumes that the whole chain of modules works properly and we are able to classify the muscles into groups.

The test gathered the statistics from the talking head via playing the scenarios previously developed for the ECAF toolkit. These scenarios incorporate recipe reading, poetry reading, showing a spectrum of expressions, etc. These scenarios compared to other test scripts are full of facial expressions.

Table 5.8: Muscle usage statistics; Muscle groups' classification

Group	Name of the muscle	Usage of the muscle
M1	Right Frontalis Major	0.008489
	Left Frontalis Major	0.012070
	Right Secondary Frontalis	0.029606
	Left Secondary Frontalis	0.031259
	Right Frontalis Inner	0.063206
M2	Right Inner Labi Nasi	0.100110
	Left Frontalis Inner	0.104379
	Left Inner Labi Nasi	0.188379
	Left Frontalis Outer	0.196276
M3	Mentalis	0.268025
	Left Lateral Corugator	0.352659
	Right Lateral Corugator	0.352659
	Right Zygomaticus Major	0.389658
M4	Left Zygomaticus Major	0.416186
	Orbus	0.447536
	Mouth	0.478365
	Left Angular Depressor	0.496330
	Right Angular Depressor	0.530084
M5	Right Frontalis Outer	0.726985
	Left Labi Nasi	0.813551
	Right Labi Nasi	1.000000

In this test, we collected nearly 84 MB of mesh deformations data and 680 kB of muscle deformation data. See the visualization of mesh deformation in Figure 4.8 and the visualization of muscle deformation in Figure 4.10.

The figures highlight that the most stressed parts of mesh are near the mouth and chin. Also, the muscles near the mouth and in the middle of the forehead are also much stressed.

5.4.2 Large Text Experiment

This experiment tried to gather statistical data by reading a large body of text. We chose the preface and the first chapter of The Three Musketeers book by Alexandre Dumas. The input data is about 37 kB of pure text. The test itself collected about 642 MB of mesh deformations data and nearly 5 MB of muscle deformation data. See Figure 5.10.

Table 5.8 shows normalized usage of each muscle and classification of muscles into five groups of muscles with nearly the same usage.

5.4.3 Timetable Application Test

The first specialized application test is dedicated to the lexicon of timetable application. List of 323 big American cities and states was chosen and 77.1 MB of mesh deformation data and about 400 kB of muscle deformation data was acquired. The results of muscle deformation are interesting because they slightly differ from the previous test, see Figure

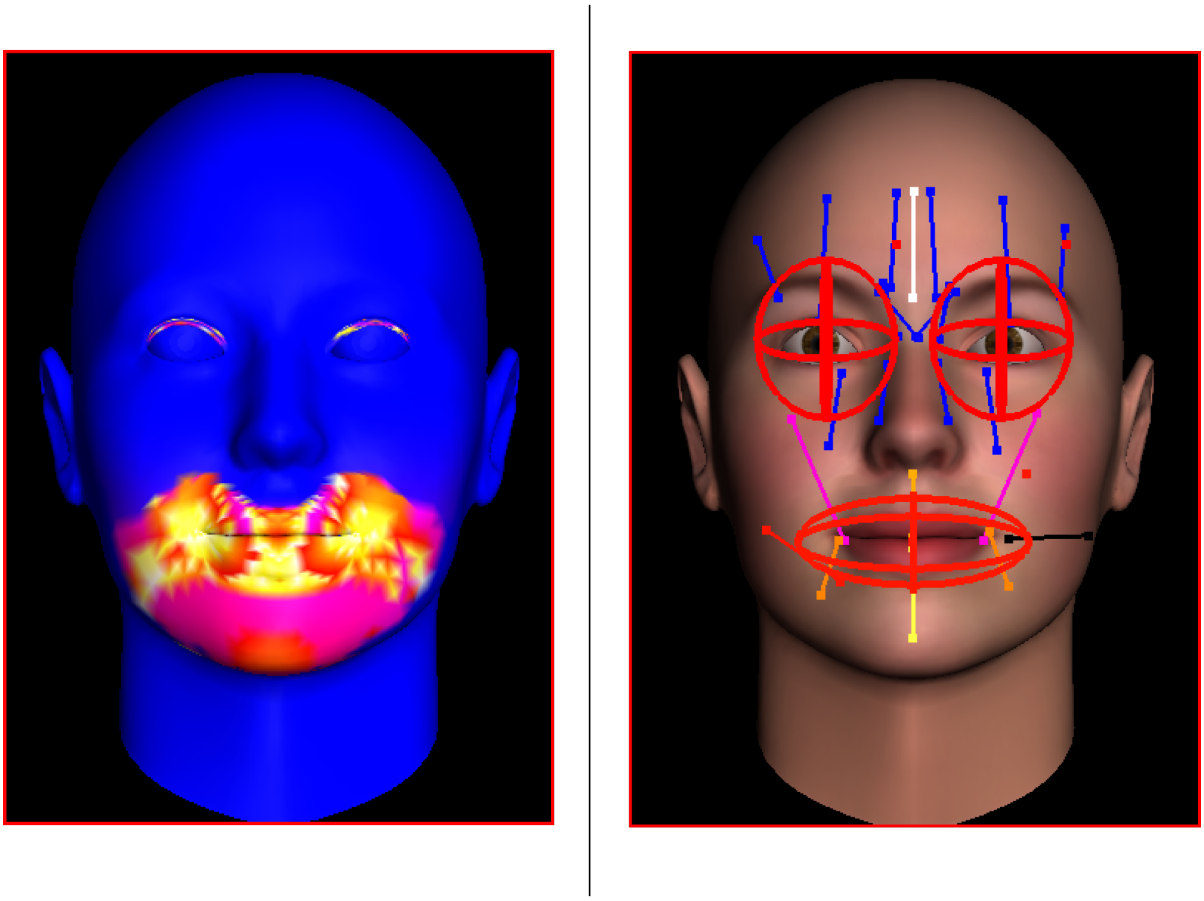


Figure 5.10: Visualization of large input text data.

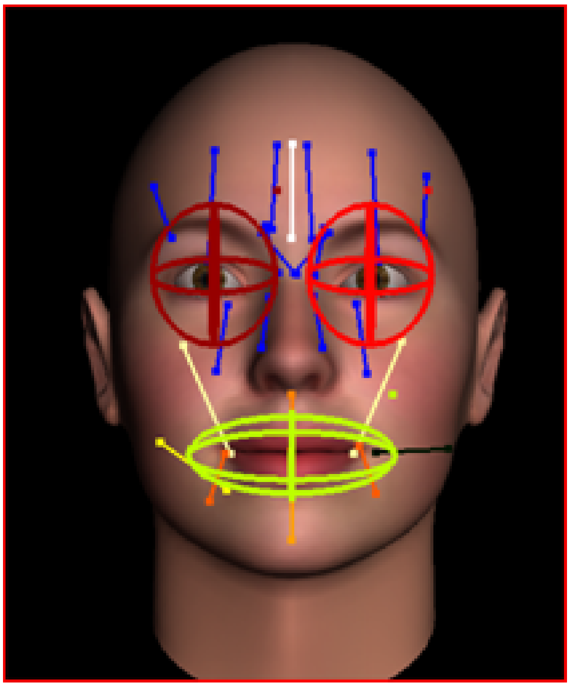


Figure 5.11: Visualization of large input text data.

5.11.

5.4.4 Weather Forecast Application Test

The last specialized test was focused on the weather forecast application. This test consists of utterances: "Say the numbers from 0 to 100", a typical weather forecast scenario and a couple of text weather forecasts from Canada. The results were nearly the same as results of the previous test (the timetable application).

5.4.5 Performance Test

The goal of this test was to prove that our level of detail method "Adaptive Reduction of Pseudo-muscles" works and earns performance benefits.

The following three scenarios that consist of English sentences enriched by facial expressions were examined. These scenarios fully exploit the usage of muscles:

1. scenario: Show up possible facial expressions.
2. scenario: The talking head tries to copy narrator movements during TV news.
3. scenario: Part of the talking head self-introduction scenario.

The test is composed of five phases. In each phase selected groups of muscles were turned off according to the statistics in Table 5.8.

1. phase: We use all the muscles.
2. phase: The group M1 is turned off.

Table 5.9: Adaptive Reduction of Pseudo-muscle performance test. It shows the lowest FPS value.

Phase	1. scenario (FPS)	2. scenario (FPS)	3. scenario (FPS)
1	34	34	35
2	37	35	37
3	39	38	39
4	42	45	43
5	45	45	47

3. phase: The groups M1 and M2 are turned off.
4. phase: The groups M1, M2, and M3 are turned off.
5. phase: The groups M1, M2, M3 and M4 are turned off.

During the experiment, one important parameter was tracked – the lowest number of frames per second (FPS) reached. Table 5.9 lists the results of the FPS measurements.

It is obvious from the table that every phase that turns off another group of muscles is faster and computationally less expensive than the preceding one. There is a difference of 17 frames per second in the third scenario between phase 1 and 5.

These results support our hypothesis that we can adaptively lower the detail of head animation by turning off certain groups of muscles.

The first experiments have shown that the users can hardly recognize the changes in the animation quality if the muscles group M1 is turned off. Even group M2 is prone to turning off without perceptible changes if the head position is not extra close to the point of view. Switching off the groups M3-M5 is possible when the distance from the observer is larger; they are noticeable.

All of the measurements and experiments were done on a PC with the following configuration: AMD Turion64 X2 1.6 GHz, 2 GB RAM, nVidia 7400 graphics card.

5.4.6 Experimental Results Discussion

The method “Adaptive Reduction of Pseudo-muscles” was implemented and evaluated by a performance test. The test proved that the classification of muscles to groups and their adaptive turning off according the collected statistical data will gain added performance of talking head rendering.

Statistical data of ECA’s head and muscles deformation from various applications was collected, such as weather forecast, large text from the book, and the timetable application. The comparison of visualizations based on this data shows some similarities, but also slight differences between the head (muscles) deformations. This only certifies the presumption that the muscle and head deformations are application specific.

5.5 Usability Study of ECA-based Who wants to be Billionaire Game

In this section the results of usability tests of the Billionaire game are presented and discussed. The testing was done with a target user group of the Billionaire Game. It was performed in Eindhoven, Netherlands from 6th to 10th December 2010 as a part of the Netcarity project. The usability testing was a collaborative effort with Ilse Bierhoff (SmartHomes, Netherlands) and Ingrid van Regteren (SVVE, Netherlands). The testing was performed on the premises of SVVE in Frederiklaan, Eindhoven.

5.5.1 Experimental Design

The participants were presented with the Billionaire game that is inspired by the TV show 'Who wants to be a millionaire'. The user is presented with a question and four options for an answer out of which only one is correct. Also the user has an option of a hint from audience and of a hint called '50:50', which halves the number of options. Each of these hints can be used only once in a single game. For interaction with the system, the user has following modalities available: mouse, touchscreen, keyboard, and voice.

The Billionaire game was presented to the participants on a PC equipped with a touchscreen. Firstly, the users were shortly introduced into the game principle and each control modality was explained. Users were encouraged to use the modality that s/he prefers, including switching modalities. At the end of the test each participant was asked to fill in a short post-test questionnaire. The Questionnaire form is attached as Appendix B.

The testing revealed very useful hints and problems. The observations are discussed in the following section.

5.5.2 User observations summary

This section tries to describe some common comments or observations that appeared during the usability tests. The following list summarizes individual observations (the U1,U2... markers mean user 1, 2...):

1. *Natural switching of modalities* (U1, U2, U3, U9, U10) – when user had difficulty with one modality s/he naturally switches to another (e.g. voice to mouse or touchscreen).

2. *Slow response means repetitions* (U1) – Reaction time of the software was sometimes slow. This feels like missing response. Therefore user repeats the utterances and this leads to a sequence of unwanted actions.

3. *User do not forgive issues for simple actions* (U1) – easily gets annoyed when the system fails to understand simple utterances like answering "yes or no".

4. *Problems with push-to-talk buttons* (U2, U10, U11, U12) – the user tends to release PTT button before finishing utterance. Or holds the PTT button too long.

5. *Human-like behavior when listening to the talking head* (U2, U12) – user tends to wait for the talking head to finish her prompt all the time. Even it was said in the introduction that s/he can interrupt her. This is mainly seen in the case of speech modality.

6. *Overarticulation* (U2, U11) – when speech recognition fails user tends to speak slower and overarticulated.

7. *Combining the disagreement with correction* (U2) – users get annoyed when repairing the answer for multiple times - uttering: “No, I did say that, this is not correct, when prompted to confirm wrong option

8. *Undo operation* (U3) – users are confused how to take back wrong answer (it’s possible in the confirmation dialog, however it is not clearly visible).

9. *Great fun with help options* (U6) – Users noticed ’50:50’ help and public help and commented this as great fun in the game.

10. *Identifying with the talking head* (U6, U11) – when commenting the game s/he refers the talking head as she. Or asking the head: “are you deaf”?

11. *Selecting the type of questions* (U7, U12, U13) – users would like to select the type of questions the talking head asks.

12. *Long confirmation prompts are rather distracting* – users are impatient and it does not work in the PC game as in TV.

5.5.3 Questionnaire Evaluation

This section summarizes user responses in a questionnaire that was presented to users during the usability study of the Billionaire game. The list of questions was as follows (Appendix B):

1. What type of interaction do you prefer?

2. How do you like the talking head during game play?

3. Do you like the computer (head) voice?

4. Was the text spoken by computer (head) clear and understandable?

5. Would you prefer the computer (head) to speak slower or faster?

6. Was the recognition accuracy sufficient for you?

7. Was it easy to find out that a recognition error has occurred?

8. Was the voice recognition fast enough?

9. Were the questions asked difficult for you?

10. Is the idea of heads moderated billionaire game interesting?

11. Was the user interface of the game clear and readable?

12. What is your overall impression of the billionaire game?

13. Would you play it from time to time?

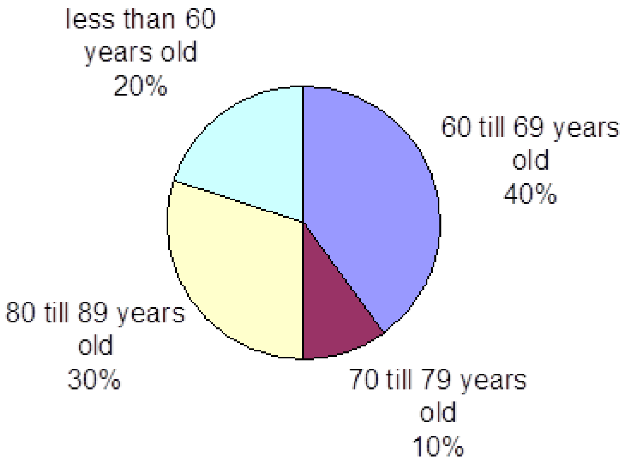


Figure 5.12: Distribution of participants' age

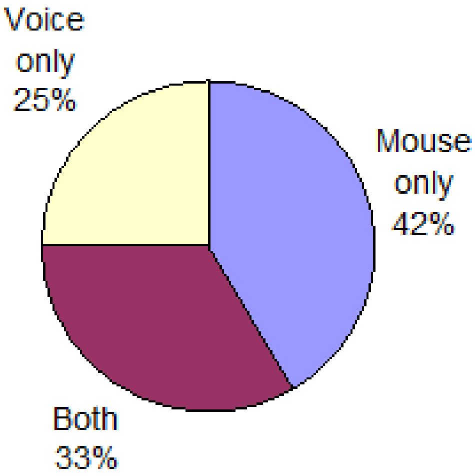


Figure 5.13: Preferred interaction modality

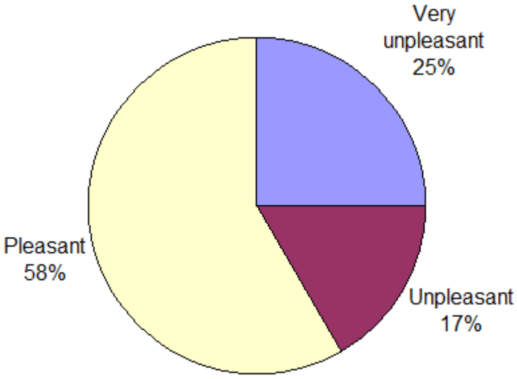


Figure 5.14: Talking head acceptance

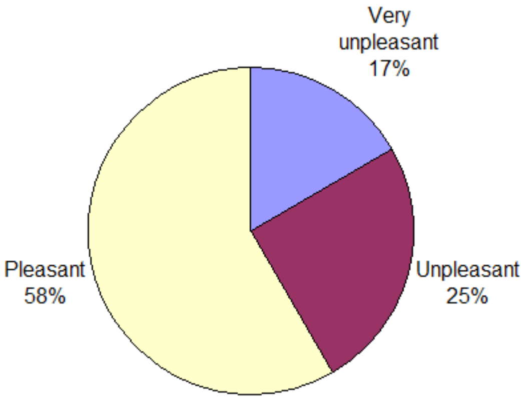


Figure 5.15: Text-to-speech voice acceptance

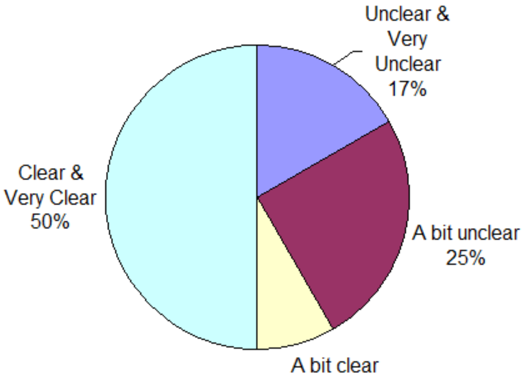


Figure 5.16: UI text legibility

The total number of participants of the usability testing was 13. The age distribution of the usability test participants is plotted in Figure 5.12.

In Figure 5.13, the preferred way of interaction with a computer of the users is shown, where the mouse was presented as the most preferred way of interaction.

The acceptance of the talking head as a user interface by the participants of the usability study was highly positive. 58% of the users found the interaction as pleasant with the rest reporting the feeling as unpleasant or worse (Figure 5.14). The same level of acceptance was reported for the text-to-speech (TTS) voice (Figure 5.15).

The legibility of the questions and answer options of the graphical user interface was reported as clear or better only by half of the users (5.16). When judging the speed of the synthesized speech by the TTS components, most of the user found the speed satisfying and approximately same proportion of users found the speed either too high or too low (see Figure 5.17.

Another interesting point was the perception of the errors in speech recognition and their influence on the fluency of the interaction. Two thirds of the users found the accuracy/success rate of the automatic speech recognition (ASR) as good or better (Figure 5.18). And at the same time spotting an error made by the ASR was reported as hard or a bit hard also by almost two thirds of the users (see Figure 5.19).

The opinion of the users about the fluency of the game and the contribution of the ASR to it is plotted in Figure 5.20, where the speed ASR was questioned. Here almost

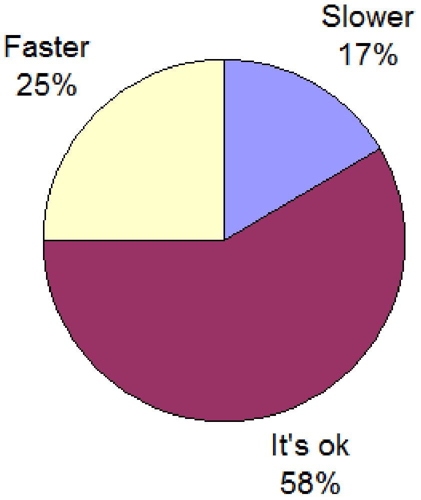


Figure 5.17: Text-to-speech system speed preference

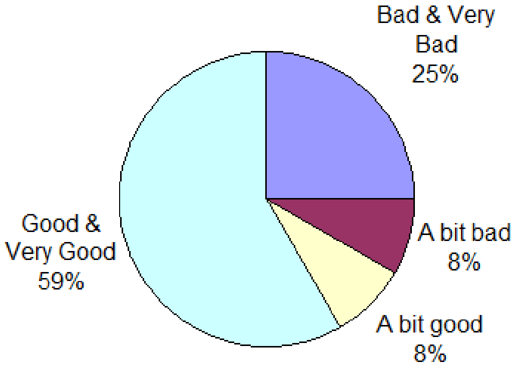


Figure 5.18: Perceived ASR accuracy

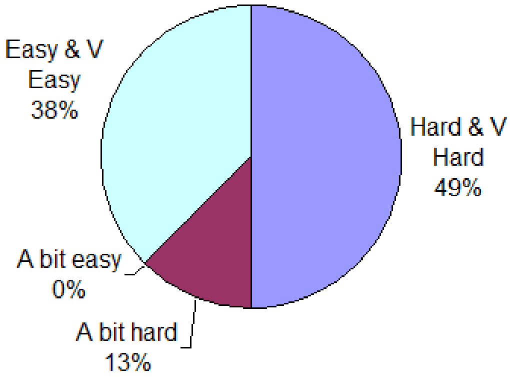


Figure 5.19: ASR error spotting

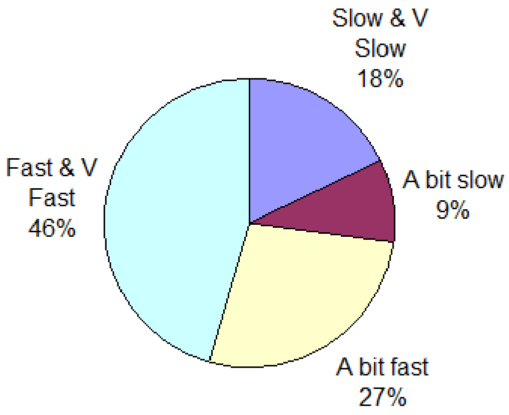


Figure 5.20: ASR speed and other game disfluencies

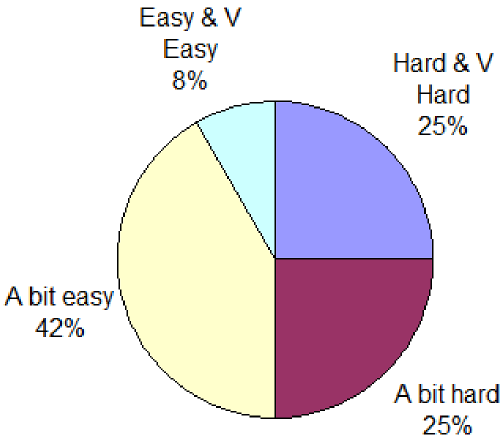


Figure 5.21: Questions difficulty level

three quarters of the users found the speed as a bit fast or better. In the interpretation, the speed of the ASR does not influence the natural flow of the game.

Further, the question difficulty was investigated (Figure 5.21). Half of the users found the questions as hard and as mentioned in the unstructured discussions with the users some even found the type of questions stereotypic.

The overall appeal of the game was highly positive (83% of user found the game interesting, Figure 5.22) and the clear form of presentation was also verified (83% of users found the user interface "clear" or better, Figure 5.23).

The overall impression with the game was clearly positive with three quarters of the users reporting pleasant or better impressions (Figure 5.24). The intention to play the Billionaire game was lower than the overall game impression with just 58% of user reporting their intention to continue playing the Billionaire game in the current form (Figure 5.25).

5.5.4 Results Discussion

The findings of usability study confirmed improvements in the game user interface from its initial version. During the usability testing the following findings need to be high-

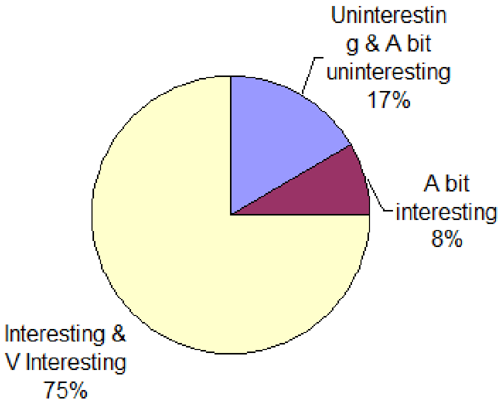


Figure 5.22: Game attractiveness

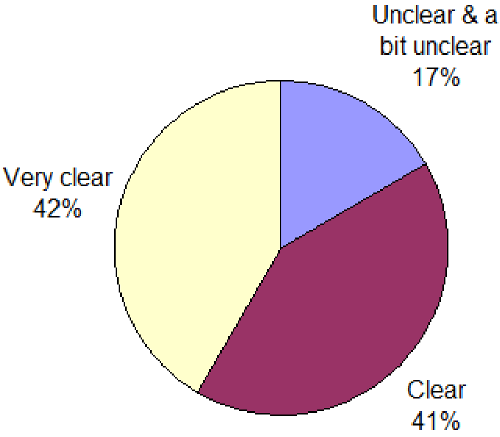


Figure 5.23: Clear user interface

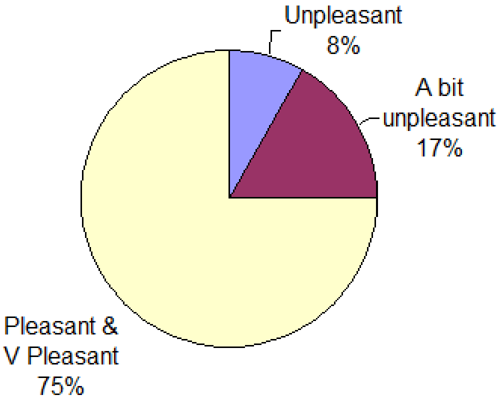


Figure 5.24: Game overall impression

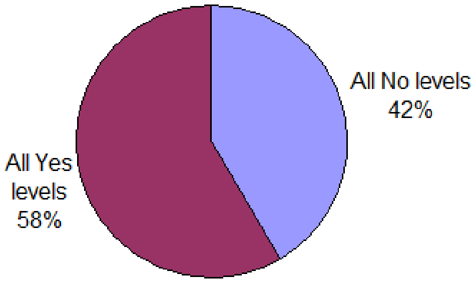


Figure 5.25: Intention to play Billionaire game

lighted:

- Although the user can switch between the multiple modalities, usually the users used a single preferred modality and switching to another one only when the currently used one stopped to work well. When this switch did not help the users got frustrated as the spectrum of modalities subjectively shrank.
- Personalization of the interaction could improve the feeling of the user from the system by using already available data (the user inputs his name in the beginning of the game).
- Talking head feels unnatural to some users, facial expressions and better articulation is seen as a potential improvement by the users.
- The structuring of the dialogs in the Billionaire game very much in line with the original TV show scenario and it seems to be disturbing during the game, mainly with the many confirmation prompts. User's did not notice the analogy with the original game for the confirmation prompts and saw them as unnecessary double checking in the game.

These points show that people like personalization when communicating with the talking head agent. The overall interaction can be vastly improved by salutation using the user name.

6 Conclusions and Future Work

This chapter summarizes the scientific advances introduced in this thesis in the area of humanized user interfaces. It lists the solutions to the issues defined in Introduction. It contains also achievements that were supported by evaluations of the ECAF talking head toolkit.

6.1 Conclusions

The main motivation of this thesis was to contribute to the research field of humanized user interfaces and to solve issues that these interfaces bring. The issues that are defined in the Chapter 1 are then presented and solved during the development process of a serious game called "Who Wants to Be a Billionaire". This game is targeted for older users. The interaction between humans and computers has many aspects. One of them is how the system/user yields or takes turns in a spoken dialog system.

ECA-based Dialog Turn-yielding

In the thesis we examined the use of turn-yielding cues in spoken dialog systems. Employment of selected visual turn-yielding cues was studied where possible and where the utilized speech synthesizer did not allow for modifying prosodic parameters for vocal turn-yielding cues in real time. We selected three vocal and two visual turn-yielding cues for examination. These cues should increase the naturalness of a conversational agent in comparison with push-to-talk initialization of speech recognition.

The main challenge is how to successfully introduce turn-yielding phenomena into an ECA-based spoken dialog system that uses concatenative speech synthesis to build a more natural system. The area of interest was narrowed to two-party (human user and a computer) conversation because of the used "Billionaire game".

Two visual cues were introduced, "talking head nod" and "stopping talking head involuntary movements". These cues are then compared to the vocal turn-yielding cues that can be found in the literature (utterance pitch, utterance final speed and utterance final loudness).

Two hypotheses were then examined by a designed perceptual experiment.

- *Hypothesis H1* – Using more turn-yielding cues before a transition relevance place increases the probability of the correct judgment about the next speaker. The turn-yielding cues can be both vocal and visual.

- *Hypothesis H2* – Visual turn-yielding cues are better than vocal cues in increasing the probability of a correct judgment of who will be the next speaker.

Two non-participating judgment experiments were performed and five visual and vocal turn-yielding cues were compared. A total of 71 participants accomplished both experiments. Each participant judged 15 dialog sequences in the first experiment or 10 dialog sequences in the second one. Findings from the first main experiment suggest that the hypothesis H1 (Using more turn-yielding cues before a transition relevance place increases the probability of correct judgment of the next speaker) is valid. Possible gender differences in the results were also analyzed. The results of statistical tests show that there are no significant differences in turn-yielding cues perception between male and female participants.

The second hypothesis (H2) that visual cues are more reliable than vocal turn-yielding cues was also validated by the first experiment. However the pitch fall cue, the head movement cue and the final speed cue results are not significant. That could be caused by more indecisive judgment results. Not to be misled by the mixture of visual and vocal turn-yielding cues, a post-test experiment was prepared. The post-test experiment validated the second hypothesis. It turned out that our selected visual turn-yielding cues were more reliable than the vocal ones (H2) in the post-test experiment. Finally, the two experiments suggest that the usage of selected visual turn-yielding cues has a positive impact on dialog turn management (better turn-ending estimates) in the area of two-party dialogs.

The findings concerning vocal turn-yielding cues are a little bit surprising. The selected vocal turn-yielding cues seem to have little impact on correct turn-ending judgment. Furthermore, the pitch fall cue had a negative effect on this judgment in the first experiment as well as in the second experiment, however, the findings were not statistically significant. These results are in contradiction to previous studies considering vocal turn-yielding cues.

The results of the experiments show clearly the advantage of ECA usage in speech dialog systems. Spoken dialog system architects have the ability to include turn-yielding cues of ECA and use state-of-the-art concatenative speech synthesis simultaneously in their systems. This could make the systems more natural and improve their interactivity. Efficiency of interaction seems to be very good even in push-to-talk systems.

Seamless Combination of "Classical" GUI Interfaces and Talking Agents

The proliferation of animated characters can be boosted by the existence of effective authoring languages and architectures supporting real-time generation of behavior based on expressive concepts. Many ECA authoring languages strictly divide the two worlds of talking agent interfaces and "classical" GUI containing buttons, texts, etc. ECA authoring languages tend to be expressive when controlling the talking agent but they are not useful when implementing a whole GUI application.

However, for users and developers the application is one piece, the GUI and the talking agent together. The authoring language should provide the means for the synchronization of the talking agent and the GUI. The proposed ECAF language builds on the concept of blended communication channels that comprise the multimodal communication acts rendered to the user at real-time. Effectiveness, practicability, and extensibility are accented as the key requisites of the ECAF language design.

Many applications have already been designed using ECAF, out of which we sketched mainly the "Billionaire game". It turns out that it is good to keep the modal blending possibilities wide (ECAF supports over 15 channels), as the application authors are to decide the proper modality mix, given the target set of users, the environment and the occasion. This is especially important concerning the basic existence of two distinct user personas: one group tends to enjoy emotional and very lively ECAs, while the other group requests presentation of facts without emotions and with a conservative behavior. The observations are clearly seen on the usability test of the "Billionaire game". The

ECAF toolkit should be able to effectively support both.

Static and Dynamic Appearance parameters

A perceptual evaluation of four “dynamic” and “static” appearance parameters was presented (eyes blinking, teeth color, head movements, mouth opening). Various parameters of appearance influence the perception of a talking agent. In total 93 participants completed our pairwise two-alternatives forced choice test. Based on their preferences each participant evaluated fifteen ECA comparison videos with different settings of examined behavior parameters.

The statistically significant results show that participants were more sensitive to some parameters (their standard scores were very different) than to others (their scores were almost the same). The behavioral parameters as teeth color, head movement and mouth opening seem to be good choices for an agent personalization. The “wizard” method of personalization is likely to be possible in the following way. The user goes through a preference pairwise video evaluation test (comparisons) and according to the results, relevant parameters will be personalized.

McGurk Test Evaluation Improvement

The McGurk effect test is a method to evaluate articulation of talking agents. This thesis addresses the issue of corrected vision influence on the McGurk effect perception. A person with corrected vision is the person that needs some vision correction, e.g. glasses. The McGurk effect shows that humans use both hearing and vision modalities in parallel to perceive and understand speech. The first experiment was presented in [115]. There dubbed videotape of visual *ga* syllable with audio *ba* syllable was presented. Most participants thought that *da* syllable was pronounced. During the talking agent articulation measurements participants are given synthetic talking agent McGurk sequences and their confusion responses are measured.

The hypothesis validated in this thesis is the following:

People with corrected vision will judge the McGurk effect differently than people with non-corrected normal vision.

In total 32 participants took part in the experiment which was designed. The participants followed the McGurk audio-video sequences using a talking head and a human head. Part of participants had corrected vision (41%), the other part of participants had normal vision. The results clearly show that there are some differences between the group of participants that had corrected-to-normal vision and participants with non-corrected normal vision. The results could help researchers that want to evaluate their developed talking agents articulation using the McGurk perception test. They should consider the vision correction means of their participants in their results and report both groups separately to be comparable.

High-level Behavior Patterns Extension of ECAF Language using Ontologies

Interaction with programs that use a talking head as an anchor is influenced by various context parameters (e.g. age of the user, experience with computer UI, nature of the user, nature of the message that needs to be conveyed, surrounding environment). A developer of a talking head powered application needs to take into account these parameters and features to deliver the best possible interaction.

The ECAF language presented in this dissertation thesis allows to control the fine-grained behavior of a talking head. However, it's not always easy and desirable to leave all tuning solely on the developer.

Therefore, a knowledge base of high-level interaction patterns is presented. The knowledge base is ontologically defined. This allows for future extensions and for using existing interaction pattern databases. Behavior patterns can be activated or deactivated in real-time during talking head usage.

Furthermore, the ontology defines behavior patterns contexts of use. User context is part of the knowledge base too. It can be gathered manually, automatically or semi-automatically from individual users.

There exist several patterns that are useful when dealing with an ECA that presents information. Proper usage of pauses in spoken text seems to be among the most useful ones. The ECA can use the effect of dramatic pause to emphasize an important part of the presented text. The patterns of pauses distribution can be different based in the context of news reading or for simple prompts.

Noisy environments are another good example that can use a knowledge base of behavior patterns. Changing confirmation strategies in noisy (simple acknowledgments) and in quiet environments (acknowledgment by repeating) can make communication more natural and pleasant.

6.2 Future Work

One course of the future work concerning talking agent turn-yielding can be the examination of other turn-yielding cues and/or in adding turn-taking cues, as well as experimenting with whole body gestures. Some other turn-yielding cues effects were already explored by Hjalmarsson [77], e.g. cue phrase – response eliciting cue had good turn-yielding results. Focus on the whole body provides the possibility of using even more turn-taking/yielding cues such as body posture, hand posture, etc. It would also be interesting to explore how judgment results develop when the ECA uses a wider range of gestures, not only speech visualization (lip-movement) and turn-yielding cues. Such gestures could soften movement differences in turn-endings and thus a user can be more confused by these gestures when detecting turn-endings. In addition, it might be interesting to examine whether the results stay the same when just a basic head or a cartoon head is used as an avatar.

Another course of future work should focus on the development of a behavior pattern database and to test this ontological database when creating real ECA applications.

Finally, testing applicability in the relatively new field of human-robot interaction, e.g. turn-yielding mechanisms when considering robot and human users is worth of testing.

7 Abbreviations and Dictionary

ASR	Automatic speech recognition
DM	Dialog manager
ECA	Embodied conversational agent
ECAF	Embodied conversational agent facade
GUI	Graphical user interface
HCI	Human-computer interaction
NLU	Natural language understanding
persona	Personas are fictional personal characters that are created by the designers to represent various potential user types within targeted product domain. Introduced by Alan Cooper [31]
TTS	Text-to-speech system
turn-taking	Action in human-to-human conversation that allows to take a turn in a dialog from a speaker by a listener
turn-yielding	Action in human-to-human conversation that allows a speaker to yield the turn to a listener (potentially many)
UCD	User centered design
UI	User interface

8 Appendix A

The list of utterances just before judgment points used in the main turn-yielding experiment:

- Dialog no. 1: Well, I must say, Algernon, that I think it is high time that Mr. Bunbury made up his mind whether he was going to live or to die.
- Dialog no. 2: I thought so. In fact, I am never wrong.
- Dialog no. 3: But your name is Ernest.
- Dialog no. 4: I have known several Jacks, and they all, without exception, were more than usually plain. Besides, Jack is a notorious domesticity for John!
- Dialog no. 5: Yes, Mr. Worthing, what have you got to say to me?
- Dialog no. 6: May I ask you then what you would advise me to do? I need hardly say I would do anything in the world to ensure Gwendolens happiness.
- Dialog no. 7: You are not quite old enough to do that.
- Dialog no. 8: In fact, now you mention the subject, I have been very bad in my own small way.
- Dialog no. 9: I dont think you will require neckties. Uncle Jack is sending you to Australia.
- Dialog no. 10: has known for a very brief space of time. The absence of old friends one can endure with equanimity. But even a momentary separation from anyone to whom one has just been introduced is almost unbearable.
- Dialog no. 11: You can go on. I am quite ready for more.
- Dialog no. 12: His death must have been extremely sudden.
- Dialog no. 13: A firm of the very highest position in their profession. Indeed I am told that one of the Mr. Markby's is occasionally to be seen at dinner parties. So far I am satisfied.
- Dialog no. 14: Untruthful! My nephew Algernon? Impossible! He is an Oxonian.

The list of utterances just before judgment points used in the post-test experiment:

- Dialog no. 1: ... right a camera shop, right, head due south ... from that just ... down for about twelve centimeters, have you got a parked van at the bottom ...
- Dialog no. 2: ... if you drew a line right across the bottom of the page it wo– ... the line would end up under yacht club, right.
- Dialog no. 3: ... the stems of the 'u' the ... vertical bits are sort of three centimeters between.

- Dialog no. 4: ... it's not it's it's a sort of two o'clock almost three o'clock ... from the allotments ... over.
- Dialog no. 5: two centimeters up from the allotments if you know what i mean, do you see what I mean, but you're.
- Dialog no. 6: o you're sort of going diagonally up that way if you know what I mean ... from ... it's about twelve or thirteen centimeters.
- Dialog no. 7: right ... going from the right-hand side of the monastery over to the left-hand side.
- Dialog no. 8: no it's it's my fault, eh right so you're up there eh go right for about three centimeters.
- Dialog no. 9: if you're underlining telephone box ... ... and back up the right-hand side up the way now I'll tell you about that in a minute.
- Dialog no. 10: ... about ... eh I would say exactly sort of ... the what's it north i don't know what's the one northwest.
- Dialog no. 11: thatched mud hut, eh you're going up just go directly north ... to the side of the east lake ... under the letter "k" "e" you know ... "k" or "e" ... in east lake.
- Dialog no. 12: well you're doing a sort of hump ... you're just going up the side of the alpine garden, right, up the left-hand side ... sorry.

9 Appendix B

Billionaire Usability Study Questionnaire

9.1 Pre-test questions

1. Participants code-name (initials + number): ....

2. Sex: Male – Female

3. Age Group:
   < 40   40 - 50   50 - 60   60 - 70   70 - 80   80 - 90   > 90

4. Computer handling skills: (none) 1   2   3   4 (excellent)

5. Previous voice recognition technologies experience: none - rare - frequent

6. Language: ....

7. Language accent: native - fluent non-native - non-native

8. Do you like new technologies (computers, robots, etc.):

   (not at all) 1   2   3   4 (enthusiast)

9.2 Post-test questions

9. What type of interaction do you prefer?
   Voice only - Mouse only - Both

10. How do you like the talking head during gameplay? (Comment)
    (not at all) 1   2   3   4 (I really like her)

11. Do you like the computer (head) voice?
    (not at all) 1   2   3   4 (I really like it)

12. Was the text spoken by computer (head) clear and understandable?
    (not at all) 1   2   3   4 (very clear)

13. Would you prefer the computer (head) to speak slower or faster?
    SLOWER - IT IS OK - FASTER

14. Was the recognition accuracy sufficient for you?
    (not at all) 1   2   3   4 (perfect)

15. Was it easy to find out that a recognition error has occurred?
    (not at all) 1   2   3   4 (very easy)

16. Was the voice recognition fast enough?
    (slow) 1   2   3   4 (fast)

17. Were the questions asked difficult for you?
    (easy) 1 2 3 4 (difficult)

18. Is the idea of heads moderated billionaire game interesting?
    (not interesting) 1 2 3 4 (very interesting)

19. Was the user interface of the game clear and readable?
    (not clear) 1 2 3 4 (very nice and clear)

20. Would you prefer more or less questions in one gameplay?
    LESS QUESTIONS - IT IS OK - MORE QUESTIONS

21. What is your overall impression of the billionaire game?
    (I dont like it) 1 2 3 4 (I really like it)

22. Would you play it from time to time?
    (I dont) 1 2 3 4 (Yes, regularly)

10 Appendix C

In this section the common and less common comments from the users are presented. The comments include authors notes from observations of the users during the test sessions.

User 1 observations. When the user has difficulty with one modality he naturally switches to another (e.g. from voice to mouse/touchscreen) When faced with slow response from the system (which feels like missing response), the user repeats his action. This ofted results in a sequence of unintended actions from the buffered user actions. The user gets easily annoyed when the system fails to work for trivial actions, like "ja, nee" recognition.

User 2 observations. The user tends to release the ESC key before he finishes his utterance To utter the selected option the user says full text on the button, eg. "A - Afrikaanse Olifant" When one modality fails to work for the user he easily/automatically switches to an alternative one. Mouse/touchscreen click fails to work, the user gets the impression he needs to wait for the talking head to finish a prompt (which is not the case in reality) When the speech recognition failed the user tried to speak slower and more articulated. When the speech recognition failed for second time for the same question the user gets rather annoyed with reactions like:

"I did not say that."; No, I did say that, this is not correct." - when prompted to confirm wrong option he did not choose.

User 3 observations. The user grasped the mouse immediately to control the system Seems to be confused when makes an error and would like to take it back (which should be possible in the confirming dialog on the screen following the question) The user gets later confused from the multitude of options available to him. User comment: "You need to know what to do [How to operate the system]"

User 4 observations. The user is very natural and successful in using any modality to operate the system The speech recognition confuses the options (B) [pron: bei] and (D) [pron: dei]

User 5 observations. This user did not want to play the Billionaire game, she said she was quite strict on how to spend her time.

User 6 observations. This user was involved in the Soprano project which has partially similar focus to support elderly in their daily activities.

The questions proved to be too difficult for this user; we used the options '50:50' and 'Publiek' Initial response to the voice interaction was "that's great fun." She was the first to notice the '50:50' help option When there was a problem with the voice control she naturally switched to mouse In the discussion about the system the user uses 'she' when talking about the talking head User comment: the user felt she talked too fast to the talking head which decreased the accuracy of recognition. User comment: the questions in the game felt like "too scientific"

User 7 observations. Difficulty reading the questions, they were too small for the user (later we learned the user forgot to take his computer glasses). He can not read neither from short distance nor from longer distance (too small font). Sometimes the mouse or touch screen do not work We asked the user to try the voice control but he tried only once and switched back to mouse/touchscreen control When the question was asked the user did not pay attention to the audio prompt and read the question on the screen (even when having difficulty reading the screen) There was again a problem with the touch screen, in an attempt to overcome this touchscreen difficulty the user tried to press harder. This accidentally work which led the user to verbalized conclusion – ”Ok, you need to press harder.” The user was not aware of the spoken prompts to the extent that he asked if he could try once more with his eyes closed. (just for short piece of the game) User comment: would like to have more personalized game, like calling him by his name, reflecting his past performance on the same question User comment: would prefer more expressive facial expressions User comment: he liked the speech interaction but was used to mouse so he sticked to mouse for interaction. User comment: head phones for people with several hearing problems User comment: would like to see addition of math/logic/reasoning questions to the range. User comment: the was not aware of the spoken prompts so he asked if could try once more with his eyes closed

User 8 observations. When the user tried the speech interface he sounded like ”Oh, that’s easy” – maybe do to the fact the other options did not work well User comment: really useful for people with limited possibilities

User 9 observations. The user thinks long before inserting his name and laughs at the formulation question The user tries the keyboard but it does not work Uses the touchscreen a lot User comment: it is like the voice is stuck in the throat of the person User comment: the confirmations are annoying, especially that he has a problem to even select an answer. User comment: likes the possibility to switch between modalities: ”If one does not work, I just try another one”.

User 10 observations. The user has problems with the keyboard input, restart of talking head and CIMA helped Very naturally started to use the speech interaction, all modalities worked fine Naturally switches between different modalities Holds the ESC key as long as the next dialog appears

User 11 observations. This is an interesting user with the age in the 70-79 range. She has just limited experience with computers

She has never used a computer before, does not know what a SPACEBAR is, we navigate She is not able to handle pressing ESC while talking, releasing after the speech is done. She presses it and keeps focusing on that activity so hard she forgets to speak. When she learns how to do it she holds the ESC key after she uttered her prompt and is wondering why is nothing happening When the computer asks again because it did not understand, she speaks louder. The speech reco works nice and the conversation is fluent She is surprised the head is confirming everything even that this is in the original Millionaire game scenario. When the system fails to understand the user easily returns to the question via the confirmation dialog Then she got into a feeling she did not speak loud enough for the reco to work. Often the user starts her line ”ja” and then she changes to the ”klaar” prompt, TRANSLATION ISSUE User comment: ”is the lady deaf?” User comment: she felt the questions were too difficult, but in general and with other questions she would like the system

User 12 observations. A user with no previous experience with a computer.

The user has real difficulties to utter her response after pressing the ESC button. Often she presses the button, holds it, searches her mind for an answer, and after releasing the ESC key she says her prompt. Errors in recognition (confusing A with C) confuse the user and his confidence in the speech recognition Several times the speech recognition stops working, microphone goes gray and there is no action from speech recognition She tried once to press the option in the text of the question She tried to use the voice in the first few questions bus as soon as she learned the touch screen works well she switched to it. User comment: the game is nice, but it would be nice to be able to choose what types of question you would like to have. User comment: she said she was really impatient, because she felt she was not doing well

User 13 observations. The user starts with mouse and clicks on the answer options before the talking head finishes her line. We proposed the user to use speech interaction:

- shy a little in the initial interaction
- then he starts with less respect, instead of 'klaar' in the confirmation dialog uses the more natural 'volgende vraag'
- uses just short prompts (one-word utterances) – 'klaar', '(A)'.
- tends to talk (even bends) in the direction of the microphone icon

The user only reads the questions, does not listen to the talking head There is sometimes bad word order in the dynamically generated prompts The long confirmation sequence gets really distracting (what work on TV does not seem to work in PC game) When using the speech inputs, she waits for the talking head to finish, when using mouse for her inputs she does not wait for her to finish. User comment: she finds the type of questions monotone, would prefer more "open-ended", thoughtful, or puzzle-like questions User comment: she dislikes the cheering from the head saying "I am not a little girl." User comment: the user has not a good feeling from the talking head, it feels not natural

11 Appendix D

This section briefly introduces the original format of “Who Wants to Be a Millionaire” TV show format which is used as inspiration for ECA-based game described in this thesis.

“Who Wants to Be a Millionaire” is a British TV quiz show. This show offers a maximum cash prize of one million pound for correct answers to multiple-choice questions. The difficulty of questions is increased during the show. The show is rather popular and spread to many other countries. Television share the same format of the show.

There is one anchor and one participant of the show in the center of studio. They are surrounded by public crowd. The anchor asks a question and offers multiple (in this case 4) answers. The participant of a game needs to select the correct one. S/he is often distracted by the anchor by questions like “Is that your final answer?” The game ends when the participant answers incorrectly. S/he can use the help of public crowd or friend on the telephone.

12 Bibliography

[1] *Oxford English Dictionary H – Definition of knowldedge.* 1989.

[2] A. H. Anderson and others. The HCRC map task corpus. *Language and Speech*, 34(4):351–366, 1992.

[3] J. R. Anderson. *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, USA, 2007.

[4] E. André and C. Pelachaud. Interacting with embodied conversational agents. *Speech technology*, pages 123–149, 2010.

[5] E. André and T. Rist. Controlling the behavior of animated presentation agents in the interface: Scripting versus instructing. *AI Magazine*, 22(4):53–66, 2001.

[6] A. Arch. Web accessibility for older users: A literature review. 2010.

[7] J. Aron. How innovative is apple's new voice assistant, siri? *New Scientist*, 212(2836):24, 2011.

[8] U. P. Association. What is user-centered design?, 2008. Available at `http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html`.

[9] A. Aubel, R. Boulic, and D. Thalmann. Real-time display of virtual humans: levels of details and impostors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(2):207–217, 2000.

[10] K. Balci, E. Not, M. Zancanaro, and F. Pianesi. Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In *Proceedings of the 15th international conference on Multimedia*, pages 1013–1016. ACM, 2007.

[11] P. Barkhuysen, E. Krahmer, and M. Swerts. The interplay between the auditory and visual modality for end-of-utterance detection. *The Journal of the Acoustical Society of America*, 123(1):354–365, 2008.

[12] J. Bates. The role of emotion in believable agents. *CACM*, 37(7):122–125, July 1994.

[13] M. Beckman and J. Hirschberg. The ToBI annotation conventions. *Ohio State University*, 1994.

[14] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. Dialog in the railtel telephone-based system. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 550–553. IEEE, 1996.

[15] D. Bohus and A. Rudnicky. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361, 2009.

[16] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[17] S. Brave, C. Nass, and K. Hutchinson. Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *International Journal of Human-Computer Studies*, 62(2):161–178, 2005.

[18] R. L. Burback. Heterogenous systems, 1998. Available at `http://infolab.stanford.edu/~burback/dadl/node95.html`.

[19] S. Card, T. Moran, and A. Newell. The model human processor. *Ariel*, 192:50–50, 1986.

[20] R. Carlson and J. Hirschberg. Cross-cultural perception of discourse phenomena. In *Interspeech 2009: 10th Annual Conference of the International Speech Communication Association*, pages 1723–1726. ISCA-INST, 2009.

[21] R. Carlson, S. Hunnicutt, and J. Gustafsson. Dialog management in the waxholm system. In *ESCA Workshop on Spoken Dialogue Systems*, pages 137–140, 1995.

[22] J. Cassell, H. Vilhjálmsson, and T. Bickmore. Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486. ACM, 2001.

[23] Cassell, Justine. *Embodied conversational agents*. 2000.

[24] C. Chao and A. L. Thomaz. Timing in multimodal turn-taking interactions: Control and analysis using timed petri nets. *Journal of Human-Robot Interaction*, 1(1):4–25, 2012.

[25] L. Chen, R. Rose, Y. Qiao, I. Kimbara, F. Parrill, H. Welji, T. X. Han, J. Tu, Z. Huang, M. P. Harper, F. K. H. Quek, Y. Xiong, D. McNeill, R. Tuttle, and T. S. Huang. VACE multimodal meeting corpus. In S. Renals and S. Bengio, editors, *MLMI*, volume 3869 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2005.

[26] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2005.

[27] N. Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

[28] N. Christoph. Empirical evaluation methodology for embodied conversational agents. *From Brows to Trust*, pages 67–90, 2005.

[29] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the fifth ACM international conference on Multimedia*, pages 31–40. ACM, 1997.

[30] G. Consortium. Gethomesafe: European commision collaborative project fp7. 2013. Available at `http://www.gethomesafe-fp7.eu/`.

[31] A. Cooper. *The Inmates are running the Asylum.* SAMS, 1999.

[32] E. Cosatto, J. Ostermann, H. P. Graf, and J. Schroeter. Lifelike talking faces for interactive services. *Proceedings of IEEE*, 91(9):1406–1429, Sept. 2003.

[33] D. Cosker, S. Paddock, D. Marshall, P. Rosin, and S. Rushton. Towards perceptually realistic talking heads: models, methods and mcgurk. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 151–157. ACM, 2004.

[34] J. Cuřín, M. Labský, T. Macek, J. Kleindienst, H. Young, A. Thyme-Gobbel, H. Quast, and L. König. Dictating and editing short texts while driving: Distraction and task completion. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 13–20. ACM, 2011.

[35] A. Cutler and M. Pearson. On the analysis of prosodic turn-taking cues. *Intonation in discourse*, pages 139–156, 1986.

[36] J. Cuřín et al. Voice-driven jukebox with ECA interface. In *Proc. of of 13th International Conference Speech and Computer*, pages 146–151, 2009.

[37] N. Dahlbäck, A. Flycht-Eriksson, A. Jönsson, and P. Qvarfordt. An architecture for multi-modal natural dialogue systems. In *ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems*, pages 53–56, 1999.

[38] H. A. David. *The method of paired comparisons*, volume 12. DTIC Document, 1963.

[39] I. de Kok and D. Heylen. Multimodal end-of-turn prediction in multi-party meetings. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 91–98. ACM, 2009.

[40] I. de Kok and D. Heylen. The multilis corpus–dealing with individual differences in nonverbal listening behavior. *Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues*, pages 362–375, 2011.

[41] E. d'Eon and D. Luebke. *GPU Gems 3 - Techniques for Realistic Real-Time Skin Rendering*, chapter 14, pages 293–348. Addison-Wesley, 2007.

[42] N. Dimakis, J. Soldatos, L. Polymenakos, P. Fleury, J. Curín, and J. Kleindienst. Integrated development of context-aware applications in smart spaces. *Pervasive Computing, IEEE*, 7(4):71–79, 2008.

[43] S. Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of personality and social psychology*, 23(2):283–292, 1972.

[44] S. Duncan and D. Fiske. *Face-to-face interaction: Research, methods, and theory.* L. Erlbaum Associates, 1977.

[45] S. J. Dyck J.L. Age differences on computer anxiety: the role of computer experience, gender and education. *Journal of Educational Computing Research*, 10:239–248, 1994.

[46] I. M. E. Andr, K. Concepcion and L. van Guilder. Autobriefer: A system for authoring narrated briefings. In O. Stock and M. Zancanaro, editors, *Multimodal Intelligent Information Presentation*, pages 143–158. Springer, 2005.

[47] W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E.-G. Schukat-Talamazzini. A spoken dialogue system for german intercity train timetable inquiries. In *In Proc. European Conf. on Speech Technology*, pages 1871–1874, 1993.

[48] J. Edlund, M. Heldner, and J. Gustafson. Utterance segmentation and turn-taking in spoken dialogue systems. *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, pages 576–587, 2005.

[49] E. Eide, A. Aaron, R. Bakis, R. Cohen, R. Donovan, W. Hamza, T. Mathes, M. Picheny, M. Polkosky, M. Smith, et al. Recent improvements to the ibm trainable speech synthesis system. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–708. IEEE, 2003.

[50] M. El-Nasr, K. Isbister, J. Ventrella, B. Aghabeigi, C. Hash, M. Erfani, J. Morie, and L. Bishko. Body buddies: social signaling through puppeteering. *Virtual and Mixed Reality-Systems and Applications*, pages 279–288, 2011.

[51] M. Eskenazi, A. Black, A. Raux, and B. Langner. Lets go lab: a platform for evaluation of spoken dialog systems with real world users. *Proceedings of Interspeech, Brisbane*, 2008.

[52] R. Fernández, T. Lucht, K. Rodríguez, and D. Schlangen. Interaction in task-oriented human-human dialogue: The effects of different turn-taking policies. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 206–209. IEEE, 2006.

[53] L. Ferrer, E. Shriberg, and A. Stolcke. Is the speaker done yet? faster and more accurate end-of-utterance detection using prosody. In *ICSLP*, pages 2061–2064, 2002.

[54] L. Ferrer, E. Shriberg, and A. Stolcke. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–608. IEEE, 2003.

[55] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer graphics: Principles and practice in C.* Addison-Wesley, 2nd edition, 1996.

[56] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3):143–166, 2003.

[57] C. Ford and S. Thompson. Interactional units in conversation: Syntactic, intonational, and pragmatic resources for the management of turns. *Studies in interactional sociolinguistics*, 13:134–184, 1996.

[58] M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, and S. Young. Training and evaluation of the his pomdp dialogue system in noise. In *Proceedings of the 9th SIGDIAL Workshop on Discourse and Dialogue*, pages 112–119. Association for Computational Linguistics, 2008.

[59] M. Gašić and S. Young. Effective handling of dialogue state in the hidden information state pomdp-based dialogue manager. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):4, 2011.

[60] P. Gebhard, M. Schröder, M. Charfuelan, C. Endres, M. Kipp, S. Pammi, M. Rumpler, and O. Türk. IDEAS4Games: Building expressive virtual characters for computer games. In H. Prendinger, J. C. Lester, and M. Ishizuka, editors, *Intelligent Virtual Agents, 8th International Conference, IVA 2008, Tokyo, Japan, September 1-3, 2008. Proceedings*, volume 5208 of *Lecture Notes in Computer Science*, pages 426–440. Springer, 2008.

[61] P. Gedalia. The expression toolkit an open-source procedural facial animation system. Available at `http://expression.sf.net`.

[62] P. Gedalia. The expression toolkit an open-source procedural facial animation system, 2013. Available at `http://expression.sf.net`.

[63] F. Goldman-Eisler. Cycle linguistics: Experiments in spontaneous speech, 1986.

[64] Google. Google now assistant, 2013. Available at `http://www.google.com/landing/now/`.

[65] A. L. Gorin, G. Riccardi, and J. H. Wright. How may I help you? *Speech Communication*, 23(1/2):113–127, 1997.

[66] M. Gorman. Development and the rights of older people. *J. Randel*, 1999.

[67] A. Gravano. *Turn-taking and affirmative cue words in task-oriented dialogue.* Columbia University, 2009.

[68] E. Gu and N. Badler. Visual attention and eye gaze during multiparty conversations with distractions. In *Intelligent Virtual Agents*, pages 193–204. Springer, 2006.

[69] C. A. Hack and C. J. Taylor. Modelling 'talking head' behaviour. In *Proc. of British Machine Vision Conference*, pages 122–132, 2003.

[70] O. Hargie, D. Dickson, and D. Tourish. *Communication skills for effective management.* Palgrave Macmillan, Basingstoke, 2004.

[71] Y. He and S. Young. Semantic processing using the hidden vector state model. *Computer speech & language*, 19(1):85–106, 2005.

[72] M. Heldner and J. Edlund. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568, 2010.

[73] A. Heloir and M. Kipp. Embr–a realtime animation engine for interactive embodied agents. In *Intelligent Virtual Agents*, pages 393–404. Springer, 2009.

[74] D. Heylen. Understanding speaker-listener interaction. 2009.

[75] D. Heylen, S. Kopp, S. C. Marsella, C. Pelachaud, and H. Vilhjálmsson. The next step towards a function markup language. In *Intelligent Virtual Agents*, pages 270–280. Springer, 2008.

[76] A. Hjalmarsson. On cueadditive effects of turn-regulating phenomena in dialogue. In *Proceedings of Diaholmia–13th Workshop on the Semantics and Pragmatics of Dialogue*, pages 27–34, 2009.

[77] A. Hjalmarsson. The additive effect of turn-taking cues in human and synthetic voice. *Speech Communication*, 53(1):23–35, 2011.

[78] Z.-W. Hong, K.-Y. Chin, and J.-M. Lin. Developing embodied agent-based user interface by using interactive drama markup language. In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pages 524–528. ACM, 2009.

[79] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–166. ACM, 1999.

[80] D. Hosmer, T. Hosmer, S. Le Cessie, and S. Lemeshow. A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in medicine*, 16(9):965–980, 1997.

[81] D. Hosmer and S. Lemeshow. *Applied logistic regression.* Wiley-Interscience, 2000.

[82] IBM. Embedded viavoice multiplatform software development kit, 2005.

[83] Jacob and R. J. K. What is the next generation of human-computer interaction? In *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems*, volume 2 of *Workshops*, pages 1707–1710, 2006.

[84] R. J. K. Jacob. Input/output devices and interaction techniques. In *In CRC Computer Science and Engineering.* CRC Press LLC: Boca, 2004.

[85] M. Jeon and B. N. Walker. Spindex (speech index) improves auditory menu acceptance and navigation performance. *ACM Transactions on Accessible Computing (TACCESS)*, 3(3):10–36, 2011.

[86] K. Jokinen. *Constructive dialogue modelling: Speech interaction and rational agents*, volume 10. Wiley.com, 2009.

[87] K. Jokinen. Non-verbal signals for turn-taking and feedback. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*, pages 2961–2967, 2010.

[88] K. Jokinen, M. Nishida, and S. Yamamoto. Eye-gaze experiments for conversation monitoring. In *Proceedings of the 3rd International Universal Communication Symposium*, pages 303–308. ACM, 2009.

[89] G. Jonsdottir and K. Thórisson. Teaching computers to conduct spoken interviews: Breaking the realtime barrier with learning. In *Intelligent Virtual Agents*, pages 446–459. Springer, 2009.

[90] G. Jonsdottir, K. Thorisson, and E. Nivel. Learning smooth, human-like turntaking in realtime dialogue. In *Intelligent Virtual Agents*, pages 162–175. Springer, 2008.

[91] D. Jurafsky and J. Martin. *Speech & Language Processing.* Pearson Education India, 2000.

[92] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, and N. Morgan. The berkeley restaurant project. In *Proc. ICSLP*, volume 94, pages 2139–2142, 1994.

[93] KDE.org. Amarok - music player, 2010. Available at `http://amarok.kde.org`.

[94] A. Kendon. Some relationships between body motion and speech. *Studies in dyadic communication*, pages 177–210, 1972.

[95] C. W. Kennedy and C. T. Camden. A new look at interruptions. *Western Journal of Communication (includes Communication Reports)*, 47(1):45–58, 1983.

[96] A. Kravchenko. 3d character model of woman – masha. 2009.

[97] M. Labskỳ, T. Macek, J. Kleindienst, H. Quast, and C. Couvreur. In-car dictation and drivers distraction: a case study. In *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments*, pages 418–425. Springer, 2011.

[98] J. Lai, C. Karat, and N. Yankelovich. Conversational speech interfaces and technologies. *Human-Computer Interaction: Design Issues, Solutions, and Applications*, pages 381–392, 2009.

[99] J. E. Laird. *The Soar cognitive architecture.* MIT Press, 2012.

[100] E. J. Langer, A. Blank, and B. Chanowitz. The mindlessness of ostensibly thoughtful action: The role of" placebic" information in interpersonal interaction. *Journal of personality and social psychology*, 36(6):635–642, 1978.

[101] K. Lee, Y. Jung, and C. Nass. Can user choice alter experimental findings in human–computer interaction?: Similarity attraction versus cognitive dissonance in social responses to synthetic speech. *Intl. Journal of Human–Computer Interaction*, 27(4):307–322, 2011.

[102] S. Lee and M. Eskenazi. Pomdp-based let's go system for spoken dialog challenge. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 61–66. IEEE, 2012.

[103] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[104] W. Lo and F. Soong. Generalized posterior probability for minimum error verification of recognized sentences. In *Proc. ICASSP*, pages 85–88, 2005.

[105] M. Louwerse, A. Graesser, D. McNamara, and S. Lu. Embodied conversational agents as conversational partners. *Applied Cognitive Psychology*, 23(9):1244–1255, 2009.

[106] M. Lusk and R. Atkinson. Varying a pedagogical agents degree of embodiment under two visual search conditions. *Applied Cognitive Psychology*, 21:747–764, 2007.

[107] K. F. MacDorman. Mortality salience and the uncanny valley. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 399–405. IEEE, 2005.

[108] M. Mancini and C. Pelachaud. The fml-apml language. In *Proceedings of the Workshop on Functional Markup Language at the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '08)*, 2008.

[109] A. Marriott, S. Beard, J. Stallo, and Q. Huynh. Vhml-directing a talking head. *Active Media Technology*, pages 90–100, 2001.

[110] D. Massaro and D. Stork. Speech recognition and sensory integration: a 240-year-old theorem helps explain how people and machines can integrate auditory and visual information to understand speech. *American Scientist*, pages 236–244, 1998.

[111] D. W. Massaro. A framework for evaluating multimodal integration by humans and a role for embodied conversational agents. In R. Sharma, T. Darrell, M. P. Harper, G. Lazzari, and M. Turk, editors, *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 2004, State College, PA, USA, October 13-15, 2004*, pages 24–31. ACM, 2004.

[112] E. McClave. Linguistic functions of head movements in the context of speech. *Journal of Pragmatics*, 32(7):855–878, 2000.

[113] S. McGlashan and others. Voice extensible markup language (voicexml) version 2.0. 2004.

[114] D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of KR 2000*, pages 483–493. Morgan Kaufmann, 2000.

[115] H. McGurk and J. MacDonald. Hearing lips and seeing voices. 1976.

[116] M. E. McTear. *Spoken Dialogue Technology: Toward the Conversational User Interface.* Springer-Verlag, Berlin, 2004.

[117] J. B. Michael Harris Cohen, James P. Giangola. *Voice User Interface Design.* Addison-Wesley, 2004.

[118] Y. Mohammad and T. Nishida. Measuring naturalness during close encounters using physiological signal processing. In *Next-Generation Applied Intelligence*, pages 281–290. Springer, 2009.

[119] C. Mortensen. *Communication theory.* Transaction Pub, 2007.

[120] S. M. Munn and J. B. Pelz. 3D point-of-regard, position and head orientation from a portable monocular video-based eye tracker. In K.-J. Räihä and A. T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*, pages 181–188. ACM, 2008.

[121] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[122] C. S. J. Nair Sankaran N., Lee Chin Chin. Older adults and attitutdes towards computers: Have they changed. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 49:154–157(4), 2005.

[123] C. Nass, J. Steuer, and E. R. Tauber. Computers are social actors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 72–78. ACM, 1994.

[124] J. Nielsen. *Usability Engineering*, chapter 1, pages 1–3. Academic Press, Cambridge, MA, 1993.

[125] R. Niewiadomski, E. Bevacqua, M. Mancini, and C. Pelachaud. Greta: an interactive expressive eca system. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1399–1400. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[126] NIST. The history of automatic recognition evaluations at nist. 2009.

[127] H. Noguchi and Y. Den. Prosody-based detection of the context of backchannel responses. In *Proc. ICSLP*, volume 98, pages 487–490, 1998.

[128] E. Not, K. Balci, F. Pianesi, and M. Zancanaro. Synthetic characters as multichannel interfaces. In *Proceedings of the 7th International Conference on Multimodal Interfaces, ICMI 2005, Trento, Italy, October 4-6, 2005*, pages 200–207. ACM, 2005.

[129] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University, 2001.

[130] Nuance. Dragon mobile assistant, 2013. Available at `http://www.dragonmobileapps.com`.

[131] Oasis. Humanml. human markup language, 2012. Available at `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=humanmarkup`.

[132] R. Ogden. Non-modal voice quality and turn-taking in finnish. *Sound patterns in interaction*, pages 29–62, 2004.

[133] M. Oliveira and T. Freitas. Intonation as a cue to turn management in telephone and face-to-face interactions. *Proceedings of the Speech Prosody 2008*, pages 485–488, 2008.

[134] N. Oliver, A. P. Pentland, and F. Berard. LAFTER: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33(8):1369–1382, Aug. 2000.

[135] B. Oreström. *Turn-taking in English conversation*, volume 66. Krieger Pub Co, 1983.

[136] M. T. Oszu and L. Ling. *Encyclopedia of Database Systems*. Springer, 2009.

[137] E. Padilha and J. Carletta. Nonverbal behaviours improving a simulation of small group discussion. In *Proceedings of the First International Nordic Symposium of Multi-modal Communication*, pages 93–105, 2003.

[138] A. Papoulis. *Probability, Random Variables, and Stocastic Processes, 2nd ed.* McGraw-Hill, New York, 1984.

[139] F. I. Parke. Computer generated animation of faces. In *Proceedings of the ACM annual conference-Volume 1*, pages 451–457. ACM, 1972.

[140] F. I. Parke. Parameterized models for facial animation. *Computer Graphics and Applications, IEEE*, 2(9):61–68, 1982.

[141] K. Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, 1985.

[142] J. Pieraccini, R. Huerta. Where do we go from here? research and commercial spoken dialog systems. In *Proc. of 6th SIGdial Workshop on Discourse and Dialog*, pages 1–10, 2005.

[143] R. Pieraccini, D. Suendermann, K. Dayanidhi, and J. Liscombe. Are we there yet? research in commercial spoken dialog systems. In *Text, Speech and Dialogue*, pages 3–13. Springer, 2009.

[144] E. W. Pieraccini R., Levin E. Spoken language dialog: Architectures and algorithms. In *Proc. of XXIIemes Journees dEtude sur la Parole*, pages ?–10, 1998.

[145] P. Piwek, B. Krenn, M. Schröder, M. Grice, S. Baumann, and H. Pirker. Rrl: A rich representation language for the description of agent behaviour in neca. In *Embodied conversation agents – let's specify and evaluate them!*, 2002.

[146] S. M. Platt and N. I. Badler. Animating facial expressions. In *ACM SIGGRAPH computer graphics*, volume 15, pages 245–252. ACM, 1981.

[147] R. Poli. Descriptive, formal and formalized ontologies. *Husserl's Logical Investigations reconsidered*, pages 183–210, 2003.

[148] R. Poppe, K. Truong, D. Reidsma, and D. Heylen. Backchannel strategies for artificial listeners. In *Intelligent Virtual Agents*, pages 146–158. Springer, 2010.

[149] G. Potamianos, J. Huang, E. Marcheret, V. Libal, R. Balchandran, M. Epstein, L. Seredi, M. Labsky, L. Ures, M. Black, et al. Far-field multimodal speech processing and conversational interaction in smart spaces. In *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, pages 119–123. IEEE, 2008.

[150] H. Prendinger, S. Descamps, and M. Ishizuka. Mpml: A markup language for controlling the behavior of life-like characters. *Journal of Visual Languages & Computing*, 15(2):183–203, 2004.

[151] J. Pruitt and J. Grudin. Personas: practice and theory. In *DUX '03: Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15, New York, NY, USA, 2003. ACM.

[152] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[153] A. Raux and M. Eskenazi. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. *Proceedings of SIGdial 2008*, pages 1–10, 2008.

[154] E. Reiter and R. Dale. *Building natural language generation systems*. Cambridge university press, 2000.

[155] J. C. Richards and R. W. Schmidt. Conversational analysis. *Language and communication*, pages 117–154, 1983.

[156] U. Ritterfeld, M. J. Cody, and P. Vorderer. *Serious games: Mechanisms and effects*. Taylor & Francis, 2009.

[157] J. Riviere, C. Adam, S. Pesty, C. Pelachaud, N. Guiraud, D. Longin, and E. Lorini. Expressive multimodal conversational acts for saiba agents. In *Intelligent virtual agents*, pages 316–323. Springer, 2011.

[158] N. Roman and A. Carvalho. Complementing rrl for dialogue summarisation. *Advances in Artificial Intelligence–IBERAMIA 2010*, pages 376–385, 2010.

[159] H. Sacks, E. A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turntaking for conversation. In A. Schenkein, editor, *Studies in the Organization of Conversational Interaction*, pages 7–55. Academic Press, New York, 1978.

[160] D. Salvucci and N. Taatgen. *The multitasking mind*. Oxford University Press, USA, 2010.

[161] D. Schaffer. The role of intonation as a cue to turn taking in conversation. *Journal of Phonetics*, 11(3):243–257, 1983.

[162] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of*

the Association for Computational Linguistics; Companion Volume, Short Papers, pages 149–152. Association for Computational Linguistics, 2007.

[163] D. Schlangen. From reaction to prediction: Experiments with computational models of turn-taking. 2006.

[164] M. Schröder. The semaine api: A component integration framework for a naturally interacting and emotionally competent embodied conversational agent. 2012.

[165] N. Sebe and A. Jaimes. Multimodal human-computer interaction: A survey. In *Computer Vision in Human-Computer Interaction*, pages 1–3, 2005.

[166] E. Selfridge, I. Arizmendi, P. Heeman, and J. Williams. Integrating incremental speech recognition and pomdp-based dialogue systems. In *Proceedings of 13th annual SIGdial Meeting on Discourse and Dialogue*, 2012.

[167] A. Sharkey and N. Sharkey. Children, the elderly, and interactive robots. *Robotics & Automation Magazine, IEEE*, 18(1):32–38, 2011.

[168] T. Shipp, K. Izdebski, and P. Morrissey. Physiologic stages of vocal reaction time. *Journal of Speech, Language and Hearing Research*, 27(2):173, 1984.

[169] B. Shneiderman and P. Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, 1997.

[170] G. Skantze. Exploring human error recovery strategies: Implications for spoken dialogue systems. *Speech Communication*, 45(3):325–341, 2005.

[171] B. Smith and C. A. Welty. FOIS introduction: Ontology - towards a new synthesis. In *FOIS*, pages iii–ix, 2001.

[172] R. Ten Ham, M. Theune, A. Heuvelman, and R. Verleur. Judging laura: Perceived qualities of a mediated human versus an embodied agent. In *Intelligent Virtual Agents*, pages 381–393. Springer, 2005.

[173] M. ter Maat and D. Heylen. Turn management or impression management? In *Intelligent Virtual Agents*, pages 467–473. Springer, 2009.

[174] M. Ter Maat, K. Truong, and D. Heylen. How turn-taking strategies influence users impressions of an agent. In *Intelligent Virtual Agents*, pages 441–453. Springer, 2010.

[175] J. Terken, H.-J. Visser, and A. Tokmakoff. Effects of speech-based vs handheld e-mailing and texting on driving performance and experience. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 21–24. ACM, 2011.

[176] D. Terzopoulos and K. Waters. Physically-based facial modelling, analysis, and animation. *The journal of visualization and computer animation*, 1(2):73–80, 1990.

[177] L. Tesnière and J. Fourquet. *Eléments de syntaxe structurale*, volume 1965. Klincksieck Paris, 1959.

[178] M. Theune. Natural language generation for dialogue: system survey. 2003.

[179] M. Theune, D. Heylen, and A. Nijholt. Generating embodied information presentations. *Multimodal Intelligent Information Presentation*, pages 47–67, 2005.

[180] K. Thórisson. Natural turn-taking needs no manual: Computational theory and model, from perception to action. *Multimodality in language and speech systems*, 19, 2002.

[181] K. Thórisson. A new constructivist ai: From manual methods to self-constructive systems. *Theoretical Foundations of Artificial General Intelligence*, pages 145–171, 2012.

[182] L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273–286, 1927.

[183] P. Turner. Crazy eddie's gui system, 2006.

[184] R. R. Vallacher and D. M. Wegner. What do people think theyre doing? action identification and human behavior. *Psychological review*, 94(1):3–15, 1987.

[185] K. Van Deemter, B. Krenn, P. Piwek, M. Klesen, M. Schröder, and S. Baumann. Fully generated scripted dialogue for embodied agents. *Artificial Intelligence*, 172(10):1219–1244, 2008.

[186] A. Waibel, H. Steusloff, R. Stiefelhagen, and K. Watson. *Computers in the human interaction loop*. Springer, 2009.

[187] M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, et al. Darpa communicator dialog travel planning systems: The june 2000 data collection. In *Proc. Eurospeech*, pages 1371–1374, 2001.

[188] N. Ward and Y. Al Bayyari. American and arab perceptions of an arabic turn-taking cue. *Journal of cross-cultural psychology*, 41(2):270–275, 2010.

[189] N. Ward, A. Rivera, K. Ward, and D. Novick. Some usability issues and research priorities in spoken dialog applications. 2005.

[190] K. Waters. A muscle model for animation three-dimensional facial expression. *ACM SIGGRAPH Computer Graphics*, 21(4):17–24, 1987.

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

[192] F. Weng, B. Yan, Z. Feng, F. Ratiu, M. Raya, B. Lathrop, A. Lien, S. Varges, R. Mishra, F. Lin, et al. Chat to your destination. In *Proc. of the 8th SIGDial workshop on Discourse and Dialogue*, pages 79–86, 2007.

[193] A. Wennerstrom and A. Siegel. Keeping the floor in multiparty conversations: Intonation, syntax, and pause. *Discourse Processes*, 36(2):77–107, 2003.

[194] J. Wiemann and M. Knapp. Turn-taking in conversations. *Journal of Communication*, 25(2):75–92, 1975.

[195] J. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[196] J. D. Williams and S. M. Witt. A comparison of dialog strategies for call routing. *International Journal of Speech Technology*, 7:9–24, 2004.

[197] M. Wong-Riley. Changes in the visual system of monocularly sutured or enucleated cats demonstrable with cytochrome oxidase histochemistry. *Brain research*, 171(1):11–28, 1979.

[198] R. S. Wright Jr, N. S. Haemel, G. Sellers, and B. Lipchak. *OpenGL SuperBible: comprehensive tutorial and reference*. Addison-Wesley, 2010.

[199] J. C. Xia, J. El-Sana, and A. Varshney. Adaptive real-time level-of-detail based rendering for polygonal models. *Visualization and Computer Graphics, IEEE Transactions on*, 3(2):171–183, 1997.

[200] J. Xiao, J. Stasko, and R. Catrambone. Embodied conversational agents as a ui paradigm: A framework for evaluation. *Embodied conversational agents-let's specify and evaluate them*, 2002.

[201] L.-c. Yang. Duration and pauses as cues to discourse boundaries in speech. In *Speech Prosody 2004, International Conference*, pages 267–270, 2004.

[202] Z. Yang and M. Ishizuka. MPML-FLASH: A multimodal presentation markup language with character agent control in flash medium. In *ICDCS Workshops*, pages 202–207. IEEE Computer Society, 2004.

[203] N. Yee, J. Bailenson, and K. Rickertsen. A meta-analysis of the impact of the inclusion and realism of human-like faces on user experiences in interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–10. ACM, 2007.

[204] V. H. Yngve. On getting a word in edgewise. In *Chicago Linguistic Society*, volume 6. 1970.

[205] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.

[206] Y. Zhang, E. C. Prakash, and E. Sung. A new physical model with multilayer architecture for facial expression animation using dynamic adaptive mesh. *Visualization and Computer Graphics, IEEE Transactions on*, 10(3):339–352, 2004.

13 Relevant Refereed Publications

Relevant Refereed Publication in Peer-reviewed Journal

[A.1] L. Kunc, Z. Míkovec and P. Slavík. Avatar and Dialog Turn-Yielding Phenomena. In *International Journal of Technology and Human Interaction (IJTHI)*, Volume 9(2), pages 66–88. IGI Global, Hershey, USA. 2013 (Ratio: 60%, 30%, 10%)

Relevant Refereed Publications in WoS

[A.2] L. Kunc, J. Kleindienst. ECAF: Authoring Language for Embodied Conversational Agents. In *10th Int. Conf. on Text, Speech and dialog, Pilsen*, LNCS 4629, pages 206–213. Springer, Berlin Heidelberg, Germany. 2007 (Ratio: 90%, 10%)

[A.3] L. Kunc, P. Slavík. Talking Head Visualizations & Level of Detail. In *International Conference Visualisation 2008, London*, pages 129–134. IEEE Computer Society, Los Alamitos, USA. 2008 (Ratio: 80%, 20%)

[A.4] L. Kunc, J. Kleindienst and P. Slavík. Talking Head as Life Blog. In *11th Int. Conf. on Text, Speech and dialog, Brno*, LNCS 5246, pages 365–372. Springer, Berlin Heidelberg, Germany. 2008 (Ratio: 70%, 15%, 15%)

[A.5] L. Kunc and P. Slavík. Study on Sensitivity to ECA Behavior Parameters. In *Intelligent Virtual Agents, Amsterdam*, LNCS 5773, pages 521–522. Springer, Berlin Heidelberg, Germany. 2009 (Ratio: 85%, 15%)

[A.6] L. Kunc and P. Slavík. Corrected Human Vision System and the McGurk Effect. In *Communications in Computer and Information Science 174*, pages 345-349. Springer, Berlin Heidelberg. 2011 (Ratio: 80%, 20%)

[A.7] M. Labský, J. Cuřín, T. Macek, J. Kleindienst, L. Kunc, H. Young, A. Thyme-Gobbel, H. Quast. Impact of word error rate on driving performance while dictating short texts. In *4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Portsmouth*, pages 179–182. ACM. 2012

[A.8] L. Kunc, T. Macek, M. Labský and J. Kleindienst. Speech-based Text Correction Patterns in Noisy Environment. In *HCI International 2013 Conference, Las Vegas*, pages 59–66. Springer. 2013

Relevant Refereed Other Publications

[A.9] J. Cuřín, J. Kleindienst, L. Kunc and M. Labský. Voice-driven Jukebox with ECA interface. In *13th International Conference "Speech and Computer" SPECOM'2009, St. Petersburg*, pages 146–151. St. Petersburg, Russia. 2009

Unrefereed Publications in WoS

[A.10] T. Macek, T. Kašparová, J. Kleindienst, L. Kunc, M. Labský, J. Vystrčil. Mostly Passive Information Delivery in a Car. In *5th International Conference on Auto-*

motive User Interfaces and Interactive Vehicular Applications, Eindhoven, to be published in 2013

Unrefereed Patent Disclosures

[A.11] T. Macek, M. Labský, L. Kunc, J. Kleindienst. %BLT% A method to provide incremental UI response based on multiple asynchronous evidence about user input Filed US Patent, 2012

[A.12] L. Kunc, M. Labský, T. Macek, J. Kleindienst. %BLT% Speech-based search using descriptive features of surrounding objects Filed US Patent, 2013

13.1 Citations

Citations of [A.2]

[A.13] J. Bech, T. Molina, E. Vilaclara, J. Lorente. Improving TV weather broadcasts with technological advancements: two cases from a 20 year perspective. In *Meteorological Applications Journal*, Volume 17(2), pages 142–148. Wiley. 2010

[A.14] M. Borzestowski, M. Trojanowicz, M. Stzalkowski. Systems and methods for generating and implementing an interactive man-machine web interface based on natural language processing and avatar virtual agent based character. US Patent 8,156,060. 2012

[A.15] J. Macek and J. Kleindienst Exercise support system for elderly: multi-sensor physiological state detection and usability testing In *Human-Computer Interaction – INTERACT 2011*, pages 81–88. Springer. 2011

[A.16] J. Danihelka, R. Hak, L. Kencl, J. Zara. 3d talking-head interface to voice-interactive services on mobile phones. In *International Journal of Mobile Human Computer Interaction*, Volume 3(2), pages 50–64. IGI Global. 2011

Citations of [A.4]

[A.17] J. Danihelka, L. Kencl, J. Zara. Reduction of animated models for embedded devices In *18th International Conference on Computer Graphics, Visualization and Computer Vision*, 2010

[A.18] J. Danihelka, R. Hak, L. Kencl, J. Zara. 3d talking-head interface to voice-interactive services on mobile phones. In *International Journal of Mobile Human Computer Interaction*, Volume 3(2), pages 50–64. IGI Global. 2011

Citations of [A.9]

[A.19] P. Piwek Next Generation Navigation and Recommendation Systems In *undergraduate course in Information Technology*, Open University in the UK. 2009