Human Handheld-Device Interaction

An Adaptive User Interface

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 8 november 2010 om 12.30 uur
door

Siska FITRIANIE

ingenieur in de informatica
professional doctorate in engineering aan de Stan Ackermans Institute,
Technische Universiteit Eindhoven
geboren te Bandung, Indonesia.

Dit proefschrift is goedgekeurd door de promotoren:

Prof. dr. H. Koppelaar
Prof. dr. drs. L. J. M. Rothkrantz

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. H. Koppelaar	Technische Universiteit Delft, promotor
Prof. dr. drs. L. J. M. Rothkrantz	Technische Universiteit Delft en Nederlandse Defensie Academie, promotor
Prof. dr. C. M. Jonker	Technische Universiteit Delft
Prof. dr. M. Novak	Technische Universiteit van Praag
Prof. Ing. V. Matoušek, CSc.	Universiteit van West Bohemia
Dr. K. C. H. M. Nieuwenhuis	D-CIS Lab/TRT-NL
Prof. dr. M. A. Neerincx	Technische Universiteit Delft en TNO

The research reported here was carried out in the Man-Machine Interaction Group in the Mediamatics Department at Delft University of Technology and was part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. http://www.icis.decis.nl

Publisher: TU Delft Mediamatica

ISBN 978-90-8570-698-4

Author email: s.fitrianie@gmail.com
Printed in the Netherlands by Wöhrmann Print Service

بِسْمِ ٱللّٰهِ ٱلرَّحْمٰنِ ٱلرَّحِيمِ - *in the name of Allah, the beneficent the merciful*

To ibu, ayah and Lavitanea

Contents

Preface

The work embodied in this thesis was born from philosophy, inspiration, love and commitment.

A central concept of this thesis is the communication that occurs between humans and between humans and their world. This concept acknowledges the role of mental models in the human mind, which are constructed during communication as responses to the internal and external aspects of communication itself. Despite the advanced technology of handheld devices, it still leaves some fundamental technology drawbacks that greatly influence the way and the content of the communication. The thesis developed an array of ideas and concepts for the understanding of human handheld-device interaction (HHI) in general using engineering and analysis work. The work started with more technology pushes that consisted of a number of loose nodes of designs, implementations and tests of prototypes. Some preliminary experiments and expert interviews were conducted adding more practical and technical aspects to the concepts. The scope of our interests was wide and almost unlimited. Different methods and techniques from artificial intelligence were explored. We expect that the collection of these prototypes can give a general idea to point in the possible direction of a universal ontology of HHI, which raises the possibility of higher order integrative meta-analysis of systems for HHI.

Prof. dr. Henk Koppelaar, my promotor, mentioned the idea of *technology landscape - to bring structure to unstructured data*, when I described the range of the thesis work for the first time. This led me to see the connection between the nodes. It was fortunate that the development of each node pursued a consistent methodology and shared a common knowledge representation structure, which could be both reused and extended. At the moment I realized these connections, they just sprang into a pipeline of a bigger system that is multimodal, context-aware and affective. In a way, the system is adaptive and can be built in an ad-hoc way. I am grateful to Henk Koppelaar for his critical thought about the work. He is the person who kept asking me about writing this thesis since the first month of my appointment to the project.

Maybe it is true that I saw all the connections later when I wrote this thesis, but I am sure it was not the case for prof. drs. dr. Leon J. M. Rothkrantz. This work was my third thesis under his supervision. He guided me dedicatedly and patiently through each node until I had a complete understanding about them. His experience and hunches were the inspiration for most of the work. His way of postulating ideas made me noticing things around me. As an example, I began to observe the way my child represented her intentions, drew her world, and played with her toys. Some methods were discovered through this observation. I would like to express my profound gratitude for the great opportunities he provided by confiding this project to me. The freedom and encouragement he gave me during the work gave me wings and motivated me to develop myself in my own pace and personality.

Unfortunately, all the work had to be put on hold for two years due to my illness. During this difficult time, the kind, generosity of family and friends has been a great help. I would like to offer my sincere thanks for their support and abiding encouragement. My special esteem for Leon and Fien Rothkrantz. I am also grateful to prof. dr. Catholijn M. Jonker, dr. Onno de Wit and Jamila Eddini, who have lightened our load by prolonging TUDelft's support during my sick leave and let me focus on recovering.

This thesis could not have been realized without direct and indirect contributions of many people. My gratitude goes to friends and colleagues: from D-CIS Lab, especially dr. Kees Nieuwenhuis; from ICIS project, especially Paul Burghardt, prof. dr. Lou Boves and Dragos - for their support and pleasant team-work; Lynn Packwood - for shaping my manuscripts into a more readable form; Alin and Dana - for their support and kindness; Zhenke - for his enormous help with the experiments; Nike - for making my life in the MMI group more lively; Iulia - for her help in some experiments; and this pertains also to Ramon, Coen, Pascal, Tjerk, Hani, Hantao, Mirela, Bogdan and Adriana. My acknowledgement also to Ruud, Bart and Toos and to Indonesian families in the Netherlands, especially Rosita, Penia, teh Diah, teh Rita and mas Umar, teh Alia, family Raams and family Kamarza - for their support and encouragement.

My infinite thanks go to my family for their love, continual prayers and encouragement: family Sabikun (ayah, ibu and Fabian), family Sudjali (bapak, ibu, in-laws), family Geldof and family Borkhataria. Most of all, I am sincerely grateful to my dearest husband, Archi Delphinanto, for his innumerable loving support and understanding during my countless hours in front of the computer for the past six years. He is always there for me unconditionally and this achievement truly is his as well. Of course, I must not forget about Lavitanea L. Delphinanto, our daughter. No words can express the blessing of having her, except: **"you are the reason of all my belief, creativity, passion and tenacity!"**

Finaly, اَلْحَمْدُ لِلَّهِ رَبِّ ٱلْعٰلَمِينَ - *all the praises and thanks be to Allah, the Lord of mankind and all that exists.*

Siska Fitrianie - Delft, 11th November 2010

Chapter 1

Introduction: Interface for Mobile Users

In which the exploration of a usable user interface for handheld devices is started. The challenge of human computer interaction in mobile use context is discussed. An adaptive user interface that is multimodal, context-aware and affective, for a more natural user interaction on handheld devices is explored. Formulated research questions are presented.

To work with a machinery system, human users have to be able to control the system and assess the state of the system. User interface is a composition of instruments the users can use to accomplish their tasks and maintain the system. It provides means to control the system (input) and to allow the system to give feedback (output). The term "interface" itself refers to a layer that connects a human that is operating a machine to the machine itself. In computer science and human-computer interaction (HCI), the interface (of a computer program) refers to the graphical, textual and auditory information the program presents to the user and the control sequences that the user employs to control the program.

1.1 Computerized User Interface

One of the basic human capabilities is unquestionably *communication*. Communication can be realized between humans, between machines or between humans and machines. Communication between humans has been studied in communication theory, and communication between machines has been focused on ontology and communication acts. Finally, communication between humans and machines deals with the design, implementation and evaluation of user interfaces, which is studied in the field of HCI. History has shown the revolution of research and development in user interfaces, from the first command-based user interfaces to the sophisticated graphical ones.

A HCI process is viewed as an exchange of information between the user and the computer. Figure 1.1 describes the information process that the computer performs during computer-human dialogue. The human user either intends to import knowledge or data to the computer or intends to inform the computer what data to retrieve from it. The computer is designed to deliver the requested information to the user. It can send status messages and ask the user for more information. The information is exchanged in both directions through a physical medium. The user of a conventional computer can retrieve information by looking at the screen or by listening to the sounds that the computer produces. In the other direction, the user sends information to the computer by typing on the keyboard or manipulating the mouse.

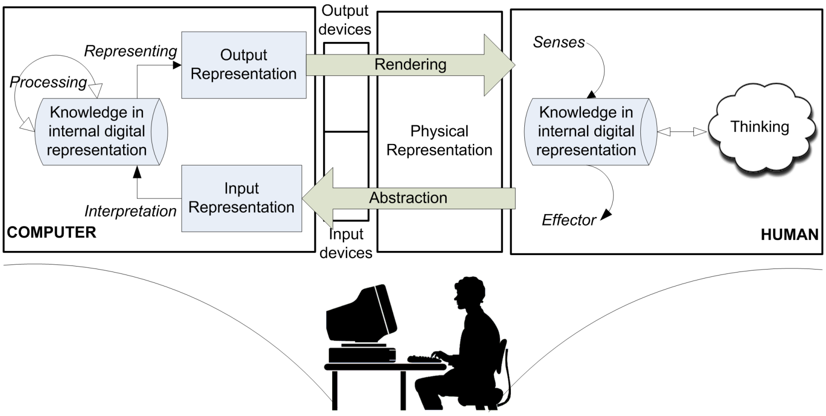


Figure 1.1: *A model of human-computer interaction*

Input/output devices form the bridge between the physical media and the digital information carried inside the computer. Typical output devices are the computer monitor for text and graphics, speakers for sounds and in a special application, or a glove for haptic output. Common input devices are keyboard and mouse. Joysticks, digital pens for handwriting and drawing, microphones, cameras and touch screens are considered as input devices that most people commonly use on a daily basis as well. The physical signals received through the input devices are highly fragmented and typically, contain no explicit semantics. An abstraction is necessary to recognize and interpret the input. Its result is a representation of concepts and can then be further processed or stored in a certain computer representation. An opposite of the transformation process is performed at the output side. The output representation is subsequently chosen and rendered into a raw description of the representation that the output devices can support. Human users perceive the information through the senses, exported through effectors. The information is internally processed and stored in distinct mental representations.

The need of a usable user interface becomes more apparent due to growing interest in balancing computation for interaction. At the same time, research and development in user interfaces is leaning toward the cognitive models of human beings parallel with the increasing complexity of the user interfaces. With the ever-increasing diffusion of computers into society, HCI is becoming increasingly essential to daily life. HCI research is focused on how computers as representatives of machines can be made easier for people to use. If the design is proper, it will make the user feel comfortable when interacting with the machine. On the other hand, even if an application is able to perform the tasks it was designed for, the application will not be satisfactory to the users unless it is able to communicate with the user in an intelligible and usable manner. The problem goes beyond how to establish an effective interface between a single computer and an individual human user. It becomes multidimensional and involves how communication between people is developing and how it should be done in an environment characterized by the ubiquitous computer. Figure 1.2 illustrates the new model for the HCI.

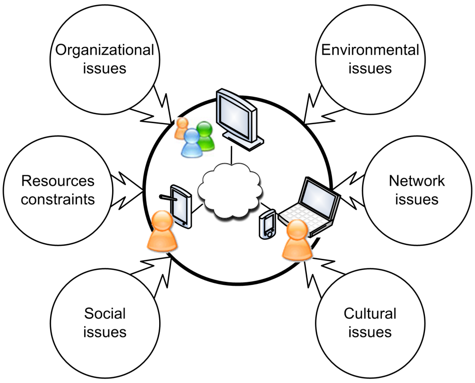


Figure 1.2: *An interaction model for ubiquitous human computer interaction*

In recent years, the concept of HCI has been extended to include not only a support for communication between humans and computers, but also communication between humans. User interface is one of the main parts in an application. Research in this field takes into account social, psychological and artistic aspects in addition to technological aspects (Nakatsu, 1998). Many different Artificial Intelligence techniques have been embedded into user interfaces to provide a more natural and productive HCI. This also includes the conveyance of context awareness (Pentland, 2000; Tennenhouse, 2000) and emotion (Pantic et al., 2006; Browne et al., 1990; Shneiderman, 2000; Picard, 1997).

1.2 User Interaction on Handheld Devices

As information and media technologies develop and become widespread, people's ability to communicate and share information increases. Recent developments in technology offer possibilities for more diverse computer types, such as workstations, notebooks and tablet computers, personal digital assistants (PDA), and smart phones. Whereas the use of a system has been traditionally seen as bounded to the workplace, current technology insists on using these devices anywhere and everywhere.

With embedded technology and connectivity, (mobile) handheld devices (Figure 1.3) are advancing to become more powerful devices. They are equipped with an increasing number of features. Word processors, personal schedulers, language programming, e-mailing, web-based services and other traditional desktop applications are increasingly available on this platform. These devices are progressively designed to be slimmer and smaller. As the terminals shrink, screen size becomes limited and keypads must either occupy less space or vanish entirely.

Figure 1.3: *(a) Sharp Zaurus, (b) Pocket PC, (c) PALM, and (d) smart-phones*

Squeezing complex functionalities into the interface of a pocket-size computer and putting it in the hand of a mobile user represents a serious interface and interaction design challenge. This challenge requires additional knowledge of HCI in a mobile use context. It is very different from the conventional systems due to:

Limited options for input. This problem refers to text input, which is difficult and time-consuming. The pen (or stylus) and touch screen duo has been widely accepted as the default hardware interfacing mechanism on handheld devices (Goldberg & Goodisman, 1991; Shneiderman, 1991). A few number of physical buttons and small-sized full-size keyboards have been available for some handheld devices. Handwriting and graffiti recognitions are also supported. However, both input methods typically require significant training. Moreover, these available input devices and methods cannot satisfy all handheld applications. For individuals who are primarily accessing stored data or inputting a small amount of entry, a keyboard is clearly not a suitable choice. Furthermore, for users trying to enter sentences into a translator system while commuting to tourism places on the street, a pen and good graffiti writing skills are simply not sufficient for the task. Recent research has been done in enhancing multimodal capabilities, such as a multimodal lifelike character (Wahlster, 2006), fusing speech and pen input (Dusan et al., 2003), speech recognition and synthesis (Comerford et al., 2001), and the like. Yet, for optimal speech recognition, the environment in which the technology is used should be the same as the training environment of the system (Bousquet-Vernhettes et al., 2003), whereas mobile activity situations are often in various environments under various conditions. The result is miss-recognition of commands, which is frustrating to the user.

Constrained screen size. This problem refers to the visualization presentation. User interface design for these devices has to cope with two opposite issues. On one side, the need to shrink the screen size to fit to the size of the device, on the other side the need to keep the screen large so that enough information can be displayed on it. Common practice and intuition suggest that users would prefer a graphical interface matched to screen size. Using a large graphic interface on a small screen would deny the user an overview of the interface at a glance. Instead, this would force the user to scroll vertically and horizontally to identify the functionality of the interface and to receive outputs from the interface. The challenge for designers is therefore to offer sufficient information and flexibility of use to users also when small displays are used. Moreover, the issues also relate to the input aspect, since according to the kind of visualization used to present information, some input devices can be more suitable than others. The problem of information navigation is part of the same concern.

Dynamic mobile use context. This problem includes unstable environment, eyes-free interaction, competition for attention resources and varying hand availability (Pascoe et al., 2000). This implies that users cannot be expected to devote their full attention to the operation of the device. Taking into account the nature of the media and its use, a handheld device offers a convenient alternative in an environment in which the circumstances make using a portable computer (for instance, laptop) inappropriate, for example when standing or walking around.

Restricted power and performance. This problem refers to a less powerful processor, less memory, and restricted power consumption, which results in poor processing power and less performance compared with a desktop computer to process information. Heavy and abundant processing work applications, such as continuous speech processing and recognition, graphics, and animation, still cannot work optimally on the current technology of handheld devices.

1.3 Research Questions

In recent years, handheld devices have been used in more complex tasks and in more dynamic environments. For example as a reporting observation system in crisis situations (Fitrianie et al., 2006, 2008b) and a mobile travel companion that helps with navigation and information retrieval in location-based services (Wahlster, 2006). In dynamic situations where task requirements and user states can change from moment to moment and the situations can be very dynamic, a real-time rapidly customizable interface between the human and the handheld technology is required to continuously maintain the best match between these entities for effective and efficient interaction.

The work in this thesis explores an adaptive user interface for (mobile) handheld devices. The research focuses, on the one hand, to exploring the ability to present input on a handheld device in ways which feel natural to the user. On the other hand, it focuses on natural output presentation to maintain interactive user system interaction. In line with these, this research focuses on investigating methods and theoretical frameworks for knowledge construction and maintenance about user, system and context to support both adaptive input and output. To be able to develop an adaptive user interface for handheld devices, one has to understand the current technology.

1. *What is the state of the art in user system interaction on handheld devices?*

Alternative input devices have been developed to support user interaction on the devices, such as pen and touch screen. Speech (Maier et al., 2006) and handwriting recognition technology (Biem, 2006) also continues to improve. Despite these developments, language input is still a bottleneck (Karlson et al., 2006). Improvement in the input method on handheld devices is still highly desired.

2. *How can we improve the performance of user input?*

Once knowledge is communicated through the input channels, it is interpreted into a coherent modality-independent representation. This representation is a combination of data structures and interpretive procedures that allows a user interface to have "knowledgeable" behavior.

3. *How can we represent the domain knowledge of a (mobile-handheld) HCI?*

Multimedia services can be generated in a coordinated fashion on a combination of multiple modality channels (Maybury & Wahlster, 1998). Coherent and synchronized multimodal representation is allocated and coordinated across media (such as text, speech and graphics). To adapt to the user's environment, the interface is designed and presented information is selected based on context variables, such as location, emotion, and available modalities. However, due to mobility and the change of environment, these contextual situations may change over time.

4. *How can we specify and produce context-sensitive and user-tailored output?*

The extension of interaction using multiple modality channels can provide users with more choices to use different modalities and find their own optimal experiences. However, the actual physical capabilities of communication devices and the type of input/output modality capabilities of the devices may differ for each user.

5. *How can we develop a multimodal interface for handheld devices that accommodates the flexible switching of communication modalities?*

To answer these questions, an experimental adaptive user interface on a handheld device has been implemented and applied to the field of crisis management. The interface allows people involved in a crisis event to report observations during the actual event. It is able to take care of the content, context and emotional loading in the user's input.

6. *Can the proposed adaptive interface support communication between human users and between the users and computers?*

It is important to stress that the aim of this project is not to replace face-to-face conversation. The developed communication interface, however, could help the users to have more control of the interaction to pursue their needs.

1.4 Research Definition: *Human Handheld-Device Interaction*

Previous research on multimodal (Oviatt et al., 2004; Reeves et al., 2004; Jaimes & Sebe, 2007), context-sensitive interfaces (Nock et al., 2004; Pantic et al., 2006) and affective computing (Hudlicka, 2003; Picard, 1997) shows that approaching the naturalness of human-human interaction plays a central role in the future of HCI and that this objective can be approached by developing cross platform HCI systems with adaptive abilities and flexible modalities suited to different environments of use. The systems are able to autonomously adjust its display and available actions to better match current goals, abilities and emotional state of the user as assessed based on user status, task and context. The design of such systems includes adequate attention to individual differences between users (such as skill, age, ability, and knowledge), supports (natural) multimodal and context sensitive interaction, and properly adapts to the emotions of the users.

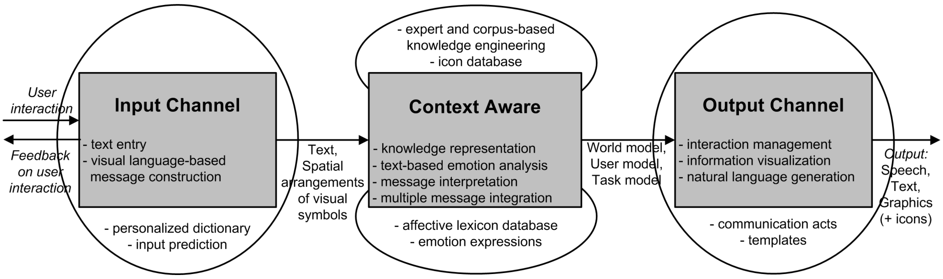

User interaction
Feedback on user interaction
Input Channel
- text entry
- visual language-based message construction
- personalized dictionary
- input prediction
Text, Spatial arrangements of visual symbols
- expert and corpus-based knowledge engineering
- icon database
Context Aware
- knowledge representation
- text-based emotion analysis
- message interpretation
- multiple message integration
- affective lexicon database
- emotion expressions
World model, User model, Task model
Output Channel
- interaction management
- information visualization
- natural language generation
- communication acts
- templates
Output: Speech, Text, Graphics (+ icons)

Figure 1.4: *The research coverage*

The research reported here more specifically aims at pursuing the same objective and approaching it by designing *an adaptive human handheld-device interaction interface framework that is also multimodal, context-aware and affective.* The thesis work spans three fields (Figure 1.4): the field of user input, of message interpretation (context-aware reasoning), and that of information generation. The types of discourse we have studied are (natural language) text and visual language-based messages. In particular, visual language-based messages are constructed using a spatial arrangement of visual symbols, such as icons, to represent concepts or ideas in mind. Text and icons are the basic, perhaps, the oldest interaction elements on any GUI-based computerized system. We consider that any research in these fields will open up opportunities to develop a more natural and robust multimodal user interface on any HCI system.

A coherent and context dependent interpretation of user messages is constructed by the employment of an ontology. The context awareness admits the analysis of user emotions from their linguistic content. With the knowledge of the user (user model), the interaction (task model) and the interpretation of user messages (world model), multimodal responses to the user system interaction that are dynamic and dependent of the observed emotion can be generated. A context-sensitive and user-tailored multimodal information presentation is produced and specified to the user's communication device. The output includes text (and speech) and visual language.

1.5 Thesis Approach

The research work primarily consists of two major threads: a fundamental conceptual thread and an engineering and analysis thread. The thesis develops the fundamental conceptual thread and subsequently strives to map the ideas of this thread to real world complex systems using the engineering and analysis thread. Here, the complex systems are humans and handheld computational devices. The study on the interaction between these complex systems enhances our understanding of complexity in general.

Table 1.1 shows an outline of the activities pursued in the research. In developing the framework, we used a user-oriented development process, which consisted of the following phases.

Requirement Analysis. In this phase, we performed two sub-phases in parallel. The first was a literature survey for a theoretical context and existing systems. The result of this sub-phase is the state of the art of the problem domain and a general model of an adaptive interface that is multimodal, context-aware and affective. Complementary to this, the second was a requirements study to identify design opportunities and additional user requirements. In this sub-phase, we performed three activities: (1) experiments on the dictionary and methods of input prediction, (2) a workshop on rationalizing the design of icons, and (3) experiments on emotion words and emotion expressions.

Design. In this phase, we developed the design concept of the framework and its components. This design follows the models set in the previous phase.

Implementation. In this phase, we developed a prototype of the research demonstrator, which was built based on the framework, under an object-oriented paradigm to support modular and reusable software development. The different features were implemented in iteration and an object class structure served as the basis for this development.

Assessment. This phase involved user testing of the prototype in a short laboratory study to address usability issues and assess the design concept. The results of the assessment were used to answer the research questions and to pose recommendations for further research and development.

Although the prototype of the communication interface has not yet been fully tested for multiple users in a real mobile-crisis setting, the user study was performed to answer whether the designed adaptive interface is usable to communicate about crisis situations. The final part of this research work has shown that the implemented interface built based on the framework satisfies the design requirements. In addition to that, the explorative study within this research has generated knowledge regarding the user requirements in the context of and the usability of adaptive interfaces on handheld devices. The rationale behind our approaches, design, empirical and analytical evaluation, and implications for research on a multimodal, adaptive and effective interface on a handheld device are described in this thesis.

Table 1.1: *The thesis research work*

No.	Action	Goal
1.	Literature survey	State of the art in human-human interaction, human-system interaction and mobile-user system interaction.
2.	Seek existing systems	State of the art and model in usability of mobile systems, adaptive user interfaces, multimodal systems and knowledge representations.
3.	Modeling	The architecture of HHI framework. The definition of the mobile context use, the adaptive interaction and modalities of the interfaces built based on the framework.
4.	Experiments on the dictionary and methods of input prediction	The characteristics of an adaptive input interface and an icon prediction.
5.	Develop and test a text-entry system	An adaptive on-screen text entry system.
6.	Develop visual language interfaces	A language independent input interface.
7.	Corpus and expert-based knowledge engineering	A methodology for developing an integrated and reusable knowledge representation for input, reasoning and output processing.
8.	Workshop on icon design	Design guidelines and considerations on designing icons and displaying icons on a visual language-based interface.
9.	Design a concept-based reasoning engine	Methods for ontology-based input interpretation and input integration.
10.	Experiments on emotion words and emotion expressions	A method for text-based emotion recognition for filtering input, detecting the user's emotion and expressing reactions toward the user.
11.	Design output channel	Methods for context-sensitive and user-tailored information presentation including language and visual generation.
12.	Develop and test a research demonstrator	Preliminary test results of proposed concepts and recommendations for future work.

1.6 Thesis Overview

The description of the research reported in this thesis starts with an introduction and theories (Chapter 1-3). These chapters describe the research on human handheld-device interaction (HHI) and the design concept of the framework. The main chapters (Chapter 4-13) describe proposed methods and approaches in forms of the modules that built up the framework. Each explanation about the modules includes background theories and experiments that have been performed in relation to the implementation of each module. The experiments may set the basis ideas on how the modules are developed or show the proof of the concepts that have brought up by the proposed modules. Before the conclusion chapter, an example of a HHI interface in the field of crisis management (Chapter 14), which was built based on the proposed framework, is presented. Similar methodology can be applied to develop other HHI interfaces in different domains.

This thesis is structured as follows:

Chapter 1 - Introduction: Interface for Mobile Users
This chapter introduces the research into a usable user interface for handheld devices. The challenge on HCI in a mobile use context is discussed. An adaptive user interface that is multimodal, context-aware and affective, for a more natural user interaction on handheld devices is explored. Formulated research questions are presented.

Chapter 2 - Interaction and Its Psychological Aspects
This chapter presents theories about communication between humans, communication between humans and systems, and communication between humans and mobile systems.

Chapter 3 - An Intelligent User Interface on Handheld Devices
This chapter introduces the model for mobile use and the model of an adaptive user interface. It also presents some surveys of recent multimodal system developments and their knowledge representation in historical perspective. Based on the study, this chapter ends by introducing a framework of an adaptive user interface that is multimodal, context-aware and affective, on handheld devices. This part of the chapter has been published in Fitrianie et al. (2010).

Part I Input Channel

Chapter 4 - Input Prediction for Adaptive Interfaces
This chapter describes two experiments. The first is to compare the methods for input prediction in general. The second is to compare the use of a common dictionary with personalized dictionaries to improve the input performance on handheld devices. This work has been published in Fitrianie & Rothkrantz (2007a). The experiment results lead to the characteristics of input prediction for an adaptive interface.

The chapter ends by proposing a prediction method for a visual language-based interface.

Chapter 5 - Text Entry with Language-based Acceleration
This chapter describes an adaptive on-screen keyboard, *the adaptive Cirrin (Circular Input)* that combines tapping-based and motion-based text input. It includes features of language-based acceleration techniques, such as a personalized and adaptive task-based dictionary, frequent character prompting, word completion, and a grammar checker with suffix completion. The chapter describes the design concept and user testing. Parts of the work have been published in Fitrianie & Rothkrantz (2007a).

Chapter 6 - Introduction to Visual Language
This chapter introduces visual language, the semiotics approach for developing visual symbols and some visual language applications. The chapter ends by introducing the motivation of the thesis work in developing a visual language-based interface.

Chapter 7 - Linguistics-based Visual Language
This chapter describes a prototype of a visual language-based interface that provides mechanisms to create messages using a sequential arrangement of visual symbols, such as icons. The description includes details of the proposed concept, the grammar and the user testing (published in Fitrianie (2004)). The chapter ends with a two-dimensional grammar for more flexible message creations (Fitrianie et al., 2006), which is based on the tree structure of a linguistic grammar.

Chapter 8 - Native Mind Visual Language
This chapter presents the latest prototype of a visual language-based interface (published in Fitrianie et al. (2008b)). The prototype allows users to create a free spatial arrangement of visual symbols using icons, lines, arrows, and ellipses. The proposed visual language was applied in the main input interface of the research demonstrator.

Part II Context-Aware Reasoning

Chapter 9 - Ontology-based Knowledge Representation
This chapter presents the knowledge representation that is utilized in the framework for input, reasoning and output processing. The knowledge representation here is focused on a specific domain, that is the crisis management field. It consists of (1) ontology of the user, the task and the world, and (2) scripts that contain scenarios about the world situations. The scripts have been developed consistently according to the concepts defined in the ontology. The research work presented in this chapter has been published in Fitrianie et al. (2006); Fitrianie & Rothkrantz (2008a)

Chapter 10 - Icon Database
This chapter presents the icon database, which has direct links to the ontology (in

Chapter 9). It also describes experiments in designing and presenting icons on a visual language-based interface. A list of design guidelines is proposed in Fitrianie & Rothkrantz (2009b).

Chapter 11 - Emotional Analysis
This chapter presents a method for analyzing emotions from the linguistic content of messages (published in Fitrianie & Rothkrantz (2008a)). It uses an affective lexicon database, which is depicted in a two-dimensional space of valence and activation. The chapter also contains the description of experiments on the lexicon database (Fitrianie & Rothkrantz, 2007b) and on human emotion expressions (Fitrianie & Rothkrantz, 2008b).

Chapter 12 - Computed Situation Awareness
This chapter presents a method for input message interpretation. The interpretation of the messages is emerged as adaptations to the outcomes of interactions of properties of reported concepts (defined by the ontology) and inferred scenarios (defined by scripts). The chapter includes the description of a method for parsing natural language into a set of concepts and of a method for integrating multiple messages using overlay operation. The method has been applied in different domains as described in Fitrianie et al. (2008b); Yang et al. (2009).

Part III Output Channel (Feedback)

Chapter 13 - Multimodal Information Generation and Presentation
This chapter presents a method for specifying and producing context-sensitive and user-tailored output. The chapter describes an interaction manager - a module for selecting communication acts and a fission output - a module for generating multimodal responses to the user-system interaction. The outputs are dynamic and dependent on the current beliefs about the user, the task and the world. The research reported in this chapter is a combination of the work that has been published in Fitrianie & Rothkrantz (2008a); Fitrianie et al. (2008a); Yang et al. (2009); Fitrianie & Rothkrantz (2009a).

Part IV Case Study and Conclusion

Chapter 14 - Case Study: Multi-User Crisis Observation Interface
This chapter presents an adaptive interface in the field of crisis management that is developed based on the proposed framework. The chapter includes the concept, design, implementation and user testing of the interface. Most of the work have been published in Fitrianie et al. (2007, 2010)

Chapter 15 - Conclusion and Recommendations
This chapter concludes the work on developing a framework of an adaptive user interface that is multimodal, context-aware and affective, on handheld devices. The chapter also points out some recommendations for future work.

Chapter 2

Interaction and its Psychological Aspects

In which the analysis of the relationship between interaction, human-human interaction, human computer interaction and mobile interaction is presented.

The term interaction can be described as a kind of *action* that occurs as two or more entities have an effect upon one another. If one of these entities is a computerized system and the other a human user then it is known as human computer interaction. Mobile human computer interaction addresses the interaction between a human user and a mobile device.

2.1 Human Human Interaction

Sperber & Wilson (1986) described interaction as a process involving two information processing devices. One device modifies the physical environment of the other. As a result, the second device constructs representations similar to the representations already stored in the first device. In human-human interaction, the information processing devices are humans and the representations are mental representations known as concepts. Concepts represent the humans themselves, the outside world and of things with which the humans are interacting (Perlovsky, 1999). These concepts provide predictive and explanatory power understanding the interaction. The results of this conceptual understanding are actions, to the outside world or inside their mind.

Interaction is understood in the entire domain of human's input and output facilities. The interaction is based on socially shared code systems, such as natural language, body language, and visual language with their own syntax, semantics and pragmatics (Wahlster, 2006). A single semiotic [1] code may be supported by many modalities. For instances, language can be supported visually (such as text and icons), aurally (such as speech) or tactilely (such as Braille scripts) and speech can have a visual component (such as lip-reading) and be supported visually (such as pointing gesture). A combination of audio, visual and other modalities in perception and the use of language being produced with a combination of vocal and physical gestures are explored. The input facilities are used to store the concepts. Here, issues of the context of language content, self-control and feedback are considered.

[1] Semiotics is the study of signs (Chandler, 2001). It concerns how signs obtain their meaning and how they convey them.

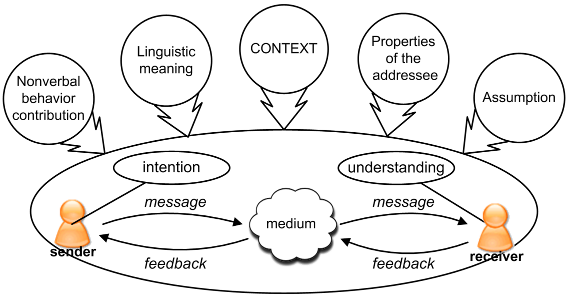


Figure 2.1: *The illustration of human-human interaction*

Meaningful interaction is trying to make another human or thing to understand what the intention is, thus creating meaning. Krauss & Fussell (1996) described how the representation stored in one individual (*sender*) is sent to, received by and constructed by the second individual (*receiver*). Figure 2.1 illustrates the sender's mental representation to be transformed into signals that can be transmitted, which in turn are transformed back into representation by the receiver. During interaction, it is often the case that the same message will be understood to mean different things in different contexts. Without defining the context, misunderstanding occurs. Moreover, even when the context is known and constant, the same message can mean different things to different addressees. The interactive use of language (in combination with non-verbal behavior) requires participants to go beyond the meaning of the literal message in extracting the sender's intended meaning. There is considerable evidence to indicate that when people design messages, they attempt to take properties of their addressees into account (Krauss & Fussell, 1991; Levelt, 1989).

During interaction, participants work collaboratively to produce shared meaning. Here, feedback is used to transform the communicated message by permitting the sender to modify tentatively formulated assumptions about what receivers know as the interaction proceeds. A speaker who is aware of the moment-to-moment state of the addressee's understanding is less dependent on a model of the addressee's knowledge constructed from prior assumptions. The speaker can avoid much of the cognitive work involved in constructing such a model. Similarly, an addressee who finds a message ambiguous or incomprehensible can avoid some of the cognitive work involved in making sense of it by signaling a lack of comprehension.

2.2 Human Multimodal Interaction

Humans use language to perform many interactive functions. Composing linguistic contents is probably the only method that can simultaneously convey a speaker's belief, intentions and meta-cognitive information about mental state along with emotional state. In written or oral conversation, language plays a role to convey information about the experiences of the utterances. The context is resulted from the process of a cognitive awareness of the current situation and the language content. In literature works, language is used to define the motivations and personalities of characters. It helps the reader to understand them and to experience vicariously their emotions.

Language is a tool to organize thinking because it bears concepts. This makes language also determines perception, which conveys forms of cognitive content since visual thinking is closely connected to verbal thought. Research in cognitive neuroscience has shown that thinking is a fusion of sensory experiences combined through the exchange of neurological signals by the brain (Solso, 1994). Thinking is a process that takes place without linguistic labels. It involves exploration, abstraction, analysis and synthesis, completion, correction, and comparison. Both external perceptions and concepts are essential for these processes, as well as for memory (Kosslyn, 1994). Damasio (1994) showed considerable evidence that the factual knowledge required for the thinking processes is presented in our minds in the form of images. Interaction using visual language, therefore, can bring more visual awareness. The thoughts could be to some degree determined by representations made available to them by such a language. This type of language can evoke a readiness to respond (Littlejohn, 2001). Hence, fast exchange of information and a fast action as a result are expected.

Humans can interact without language or in combination with nonverbal communication. As they speak, people often gesture, nod, change their postures and show facial expressions, redirect the focus of their gaze and alter their tone of voice. Although these behaviors are not linguistic by a strict definition of that term, their close combination with the speech they accompany suggests that they are relevant. They also can occur apart from the context of speech. Such body activities give real substance to face-to-face interaction in real life conversation. Mehrebian indicated that about 7% of the meaning of a message is communicated through explicit verbal channels, about 38% is communicated by paralanguage, which is basically the use of the voice, and about 55 % comes through the nonverbal channel, which includes gesture, posture and facial expressions. Allowing all of the modalities to refer to, depend upon and support each other is a key to the richness of multimodal interaction.

Emotions play an important role in human-human interaction. They are part of communication and control systems within the brain that mobilize resources to accomplish the goals specified by our motives. The instantaneous emotional state is directly linked with the displayed expression (Figure 2.2) (Ekman, 1999), which have three major functions: (1) they contribute to the activation and regulation of emotion experiences; (2) they communicate internal states and intentions to others; and (3) they activate emotion in others, a process that can help account for empathy and altruistic behavior. Such expressions do not occur randomly, but are synchronized to one's own speech or to the speech of others (Condon & Osgton, 1971; Ekman, 1979).

Humans are used to convey the thought through the (conscious or unconscious)

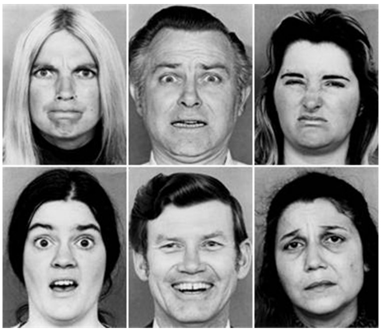


Figure 2.2: *The facial expressions of six basic emotions (Ekman, 1999): anger, fear, disgust, surprise, joy and sadness*

choice of words. Some words possess emotive meaning together with their descriptive meaning. Their meanings along with a sentence structure informs the interpretation of a nonverbal behavior and vice versa. Communicate feeling and intentions, however, cannot be mapped onto word strings in a one-to-one fashion. Rather, speakers usually select form a variety of potential alternative formulations, the ones that most exhibitory and appropriately express the meanings they want to convey. As a result, for the addressee, interpreting the literal meaning of a message is only a first step in the process of comprehension. An additional step of inference is required to derive the intention that underlies it. Seeing faces, interpreting their expression, and understanding the linguistics contents of speech are all part of human-human interaction (Pelachaud & Bilvi, 2003).

Humans are quite successful at conveying ideas to each other and reacting appropriately. This is due to many factors: the richness of the language they share, the common understanding of how the world works, and an implicit understanding of everyday situations (Abowd et al., 1999). Humans are able to use implicit situational information (context) to increase the conversational bandwidth. Unfortunately, this ability does not transfer well to humans interacting with computers. In traditional interactive computing, users have an impoverished mechanism for providing input to computers. Consequently, computers are not currently enabled to take full advantage of the context of the HCI.

2.3 Human Computer Interaction

Traditionally, HCI is viewed as any communication between the user and a computer, direct or indirect (Dix et al., 2004). The user is interacting with the computer in order to accomplish something. Direct interaction involves a dialogue with feedback and control throughout performance of a task. Indirect interaction may involve batch processing or intelligent sensors controlling the environment. Therefore, in the past, the base of concept of HCI has been researched mainly in the field of engineering. Engineers have traditionally been focusing their research on robots, agents, and other entities that feature a function for communicating with humans. Most part of the research has concentrated on the language component of communication, such as speech recognition system.

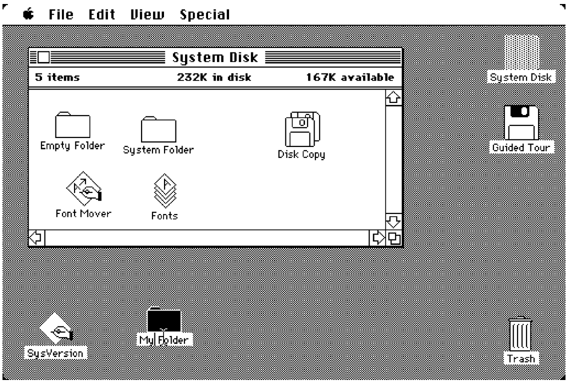


Figure 2.3: *GUI of original 1984 Macintosh desktop*

A computerized user interface is the aggregate of means by which the users interact with a particular computer program or other complex systems. A graphical user interface (GUI) is a type of interface that presents a visual fronts to a computer software that links the user to the internal working of the software. It simplifies the way the user interacts with the computer. The most common elements in GUIs are the windows, icons, menus and pointing devices (WIMP), which also denoting a style of interaction using these elements. WIMP was developed at Xeroc PARC in 1973 and popularized by the Macintosh computer (Figure 2.3) (van Dam, 1997).

Interaction with WIMP uses a physical input device to control the position of a cursor and presents information organized in windows and represented with icons. Available commands are compiled together in menus and executed through the pointing device. This reduces the cognitive load to remember and learn the possible available functions. It offers ease of use for non-technical people and novice users by informing the state of the interaction and designing high consistency between interfaces. Such generality makes the interfaces very suitable for multitasking work environments. In personal computers these elements are modeled through a desktop metaphor upon which documents and folders of documents can be placed. User interfaces for which WIMP is not well suited may use other interaction modalities, for example speech, pointing and gestures in 2D or 3D, handwriting, eye movements, facial expressions, keyboard and mouse input. Example output modalities are recorded or synthesized speech, non-speech sounds, written text, graphs, maps, tables or embodied characters that use gestures and facial expressions.

The visual front of computer software that are composed of two of more modalities are called multimodal (Bernsen, 1997). For example, charts are usually multimodal as they are composed of a graph and textual annotations. Systems that use more than one channel to communicate information are called multimodal systems. Multimodal systems use a higher level of abstraction from which they generate output and to which they transform the user input (Coutaz et al., 1993). Such systems can render the same information through different output channels and that they can fuse the input that was transmitted through multiple channels into a single message.

Wilson & Oliver (2005) uses the terms explicit and implicit interaction. An explicit interaction is where the user initiates a discrete action and expects a timely discrete response; while implicit interaction may use passive monitoring of the user over longer periods of time and result in changing some aspect of the rest of the interaction. These definitions distinguish an adaptive system from other systems. Such a system is able to adjust its interface and behavior according to the situation of the user and where the user is not aware what triggers this adaptation.

The idea of a computer "sensing" a user's emotion is coming from a vision that if a computer could recognize user emotion, the interaction will become more natural and efficient (Pantic et al., 2006; Shneiderman, 2000; Picard, 1997). Psychologists found that if a computer is able to display some anthropomorphic cues, people interact with computers in an essentially social way (Reeves & Nass, 1998). The computer could help and assist a confused user, try to cheer up a frustrated user or even empathize with the user's situation. In situations when a user may show strong emotional responses (resulting from stressful condition), such as in crisis events, the computer can assist the user to set priorities and make decisions.

Detecting emotional information begins with passive sensors that capture data about the user's physical state or behavior without interpreting the input. The data is gathered analogous to the cues humans use to perceive emotions. For example, a video camera can capture facial expressions, body posture and gestures, a microphone can capture speech and a parser can extract the choice of words in sentences. Other sensors detect the cues by directly measuring physiological data, such as skin temperature and heartbeat. Recognizing emotional information requires extraction of meaningful patterns from the collected data. Designing of computational systems to exhibit either innate emotional capabilities or that are convincingly simulate emotions is a branch research area in HCI. Some research develop lifelike characters with affective speech, facial expressions and body gestures (Cassell, 2000). Others use affective text and simple *emoticon* as facial display of emotions (Fitrianie et al., 2003).

2.4 Mobile-Human Computer Interaction

Mobile HCI can be defined as the study of the interaction between people and mobile computing systems and applications that they use on a daily basis (Love, 2005). Typically, mobile is used as an attribute of a computing device. It implies that a device can be easily transported to a location where the user wants to interact with it. Mobile computing platforms combining small, lightweight and low-power devices with wireless network connectivity enable the performance of familiar tasks in new environments. This creates opportunities for novel interactions and changes the ways people interact with computers. Many types of fieldwork that had not been previously assisted by computers can benefit from instantly available computational and informational resources. Furthermore, the connected mobile world opens up numerous possibilities beyond the realm of work expanding our communication activities.

Mobile HCI is concerned with understanding the users, their capabilities and requirements, and how these can be taken into considerations in the design of systems or applications. User experiences in the mobile use context are dramatically different from those in the traditional computing environment. They present a number of technical, environmental and social challenges in the usability of mobile devices and applications. Some challenges relate to network connectivity, which are dealing with an evolving infrastructure, coverage and feedback concerns, security hazards, and complex integration issues for a wide variety of devices. Others are posed by device design constraints, such as trade-offs between size and functionality or between weight and battery life.

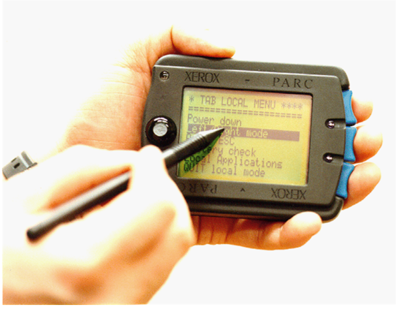


Figure 2.4: *Xeroc ParcTab, the first handheld device*

Envisioned in the beginning of 1990s that *ubiquitous computing*, intelligent small-scale technology embedded in the physical environment, would provide useful services in everyday context of people, without disturbing the natural flow of human activities (Weiser, 1991). While Bell & Gray (1997) looked back over 50 years of computing to 1947. They forecasted the next 50 years that the computers become smaller and have ability to connect to the physical world. Want et al. (1995) developed the first hand-sized computer terminal, ParcTab (Figure 2.4), to study some key technologies of ubicomputing namely PDA-devices, wireless communication and applications for the future office. Figure 2.5 presents the five major types of computer technologies in existence today, with degree of device mobility increasing from left to right based on

Weiss (2002). The focus of this thesis is user interaction on handheld devices.

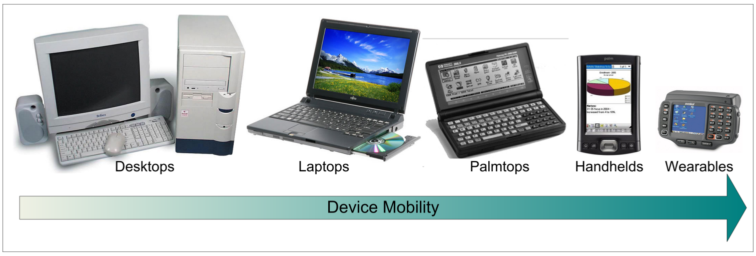


Figure 2.5: *Device mobility continuum*

Handheld devices, such as PDAs and smart-phones, are small and lightweight. They are best operated while held in the user's hand. According to Weiss (2002) a computer must pass three tests to qualify as a handheld: (1) it must be easily used while in one's hands, not resting on a table, (2) it must operate without cables, except temporarily, while recharging or synchronizing, and (3) it must either allow the addition of new applications or support Internet connectivity. Similar to laptops and palmtops, handheld devices can be easily relocated. Unlike palmtops, however, handheld devices do not require surface support outside the user's body nor does the user need to remain in one location during the interaction.

Handheld devices can be classified as fully mobile, since both the user and the device can change location while the user interacts with the device. They afford their users a mode of interaction fundamentally different from the stationary mode. Such a mode is called mobile interaction. With these devices, the freedom of movement is simply about the ability of the user to walk while using the device. Some handheld devices permit one-hand device operation. Others extend the freedom of movement to the whole body, including hands-free interaction and, in some cases, eyes-free interaction. Eyes-free mode is the ultimate in freedom of movement during interactions, as interaction requiring visual attention still constrains free body movement.

2.4.1 Mobile Interaction Characteristics

The characteristic feature of the mobile interaction is that it typically takes place while being away from the desktop and where various degrees of body movement are allowed. Mobile users face a whole new world of environmental and cognitive challenges that effect usability of devices and applications. The usability in a mobile environment is influenced both by the effects of mobile interaction and by the nature of applications running on the mobile devices.

Kristoffersen & Ljungberg (1999) point out four important features of mobile interaction: (1) task hierarchy, (2) visual attention, (3) hand manipulations, and (4) mobility. Table 2.1 contrasts and compares the features of mobile interactions (the mobile

Table 2.1: *Comparative characteristics of stationary and mobile interactions**

Interaction Parameters	Stationary Interactions	Mobile Interaction	
		Mobile Office Context	**Field Context**
Environment	Largely indoor, few fluctuations in the environment	indoor and outdoor, with frequent fluctuations	indoor and outdoor, with frequent fluctuations
Device Size	Medium to large	Small	Small
Time of Interaction	Medium to long	Short to medium	Short to medium
User Mobility	Fixed, mainly sitting position, restricted body maneuver	Any position, various degrees of free body movement allowed	Any position, various degrees of free body movement allowed
Competition for Attention	Little	Some	Significant
Task Hierarchy	Interaction-related tasks are the primary activity	Interaction-related tasks may be a secondary activity	Interaction-related tasks are mainly a secondary activity
Parallel Manipulation of Physical Objects outside Interaction	Rare	Occasional	Frequent
Interaction Styles	High dependence on direct manipulation; other styles are complementary	Greater reliance on forms and menu selection, supported by direct manipulation and natural language	Natural language is of prime importance, supplemented by menu selection and forms

*) Modified from Gorlenko & Merrick (2003).

office context and the field context) with stationary interaction (Gorlenko & Merrick, 2003). In the mobile office context, traditional office-type computing is made mobile. The assumption is that the user performs familiar work in circumstances that are less familiar for that type of work with the mobile devices acting as auxiliary devices to the stationary ones. In the field context, traditional computing has not been applied and mobile devices are the only computing devices used. The field context covers a broad range of tasks and occupations, such as service engineering, law enforcement, medical field, social work, and surveying. It also includes nonprofessional activities for which computers have not been previously used, for example shopping and travel. The tasks often replace the use other traditional medium, such as pen and paper or telephone. For mobile interaction, the values of certain parameters, such as the environment, device size, time of interaction, and user mobility, are the same for the mobile office and field contexts. The values of other parameters, such as competition for attention, task hierarchy, parallel manipulation of other physical objects, and interaction style, vary not only between stationary and mobile interactions, but also between mobile-office and field contexts.

Handheld devices with touch screen, keyboard and pen input have been promoted for field use. Recent research has been focused on their input and output capabilities and techniques and how these can be designed or used to build usable and effective user interface (Jones & Marsden, 2006; Love, 2005; Weiss, 2002). In most applications, most afford interaction with a device through either natural language or a size-able text-based input. These applications are mainly based on interaction styles involving direct manipulation, menu selection, and the use of forms (Shneiderman, 1986). WIMP are utilized with different unifying metaphors, due to constraints in space and available input devices. In most cases when text needs to be entered by hand the interaction becomes stationary, as either both hands are engaged (with a typical keyboard) or more slowly with one hand (via a pen or a chord keyboard).

Handheld devices are desirable in all field contexts, and essential in some. For example, in the area of public and emergency services. Officers can see emergency calls on their mobile devices along with the map of the area and the operator's instructions or match fingerprints of an offender against the police database (USAToday, 2001). With mobile devices, an emergency crew can report the patient's condition to the hospital, where the doctors can monitor it, advice the crew and prepare the treatment immediately after the patient arrives at the hospital. The benefits of such services to aid in life-threatening situations can be massive. In these types of activities, instantly and effortlessly interaction with the mobile devices is a must.

2.4.2 Mobile Context Awareness

Early research in context awareness of ubiquitous applications can be characterized as an attempt in finding universal context attributes for many and all such applications (Dey et al., 2001; Schmidt et al., 2001). However, mobile context encompasses larger geographical area for movement. It involves many unidentifiable users and devices, loosely defined oriented tasks, dynamic environment, interruptions and disrupting stimuli (Pascoe et al., 2000; Kristoffersen & Ljungberg, 1999). Different applications call for different context awareness. Understanding the difference between and adapting the services based on the context will help to deliver usable applications when they are needed and in the way the user wants them. In general, the contextual awareness can be categorized into the following (Gorlenko & Merrick, 2003).

Location awareness. The ability to track the user's whereabouts at each moment and provide this user with the information relevant to the current location. It is applied in location-based services, for example for offering maps and road guidance, supplying information about specific objects and places close by, or flagging the presence of other users in the area, and the like. The interpretations of location can be absolute location or relative location. In the absolute context, the user's location consists of his or her geographic or spatial coordinates at each moment in time. In the relative context, the user's location is linked to another entity (moving or stationary) for example, a car or a building.

Identity awareness. The ability to identify the user's identity.

Environmental awareness. The ability to read the specifics of the interaction setting, such as a noisy area, a one-to-one conversation or an enclosed space.

Mobility awareness. The ability to decode a user's movements and body posture at each moment.

Health awareness. The ability to measure various physical conditions of the user, such as heart rate, body temperature, and blood pressure.

Activity awareness. The ability to understand current high-level activities of the user, for example, reading, watching TV, or writing.

Emotion awareness. The ability to understand the current emotional state of the user. For some services, such as playing music and offering a specific product, knowing mood, personality and emotions of the user may become necessary.

A mobile computer is prone to enormous variations in the environment and social contexts in which it operates. An adaptive system can collect the user's traces, learn them, and use the knowledge to adjust the interface and available services to better match current context. The user may interact with the device differently, depending on the situation at hand. The system can be designed to perceive both internal and external contexts the way the user perceives them and respond to them the way the user reacts.

2.4.3 Mobile Usability Implication

The mobility has been seen as an essential requirement for shared resources and for communication with other people (Bellotti & Bly, 1996). The ability to monitor a user's whereabouts creates an opportunity for a great number of convenient context-aware services and is particularly valuable in emergency situations. At the same time, some users will consider revealing their location a serious infringement of their privacy. The social context suppresses and triggers by norms, roles and social pressure that are held in user environment (Nelson et al., 2001). For example, in face-to-face conversation, it is considered polite to keep eyes contact with the speaker instead of operating own device. Other social challenges include personalization, comfort, acceptance and trust (Demers, 1994; Chen & Kotz, 2000; Abowd et al., 1999; Henricksen et al., 2002).

Due to the mobility, often, there is no place to put the device down when there is a need to do something else. When hands are taken by another task, mobile interaction becomes impossible. Moreover, visual attention is reserved to a large extend for mobility, not interaction. The burden of environmental, such as the work context, weather conditions, physiological limitations of the human body and cognitive restrictions, will continue to put the usability of mobile devices or applications to a serious test. These create usability implications of mobile complexities that are particularly important to application designers and developers. The implications present serious challenges in the design methodology especially in (1) task analysis, (2) prototyping and (3) design evaluation (Gorlenko & Merrick, 2003).

Task analysis. For a mobile product, task analysis is significantly more complex than such analysis for a non-mobile product due to the variability of the usage environments and affects the course of task analysis in specific settings. For example, in crisis respond and rescue activities may include crowded and noisy environment. Other challenges are:

- The nature of and differences among applications that are suitable for the mobile context. Applications, that demand intense concentration, extremely high visual attention and very accurate manual input, are unsuitable for the mobile environment. A robust well-thought-out field application would support both visual and audio output modalities, as well as multimodal and manual input.

- Possibility usages of the application in different mobile settings. The users may quickly get used to the power of the product in one context and assume that it should cope with another context with equal success.

- The multitasking nature of mobile interaction. The parallel tasks may be as simple as following directions while walking or as complex as operating in a hazardous environment and applying the appropriate level of attention to both the mobile-based part of the task and whatever work it supports.

Prototyping. A prototype for mobile products will need to have a high degree of fidelity and exhibit the key characteristics of the finished product for some of the evaluations, for example, input speed performance. Successful testing of a prototype can only be performed in conjunction with a realistic simulation of the primary task. This may significantly increase the cost of prototyping.

Design evaluation. For mobile applications, design evaluation will share as most of the same difficulties that were encountered during task analysis. In particular, designers need to evaluate a mobile product in a realistic environment, where the realism may include different periods during a day, different lighting and noise levels, or even different emotional effect. Evaluating a mobile application for firefighters on fire fighting activities, where the firefighters may wear gloves or the device may not function properly due to heat conditions, tells us about the use characteristics of the application more than any other environment. Mobility, however, makes sustainable observation of a mobile product more difficult than that of a non-mobile product because it introduces far more factors to record and evaluate. Finally, the connected nature of mobile products and the flexibility of the users in the mobile environment intensely stimulate collaborative work. HCI is difficult enough for non-mobile collaborative applications, and it will certainly be more difficult for mobile collaboration.

Chapter 3

An Intelligent User Interface on Handheld Devices

In which models and approaches on mobile use context, adaptive user interface and multimodal user interface that influence the design of a usable user interface on (mobile) handheld devices, are discussed. A framework of an adaptive user interface on handheld devices is proposed.

Although hardware of handheld devices is steadily improved, the main limitations are still the same: (1) lack of screen size/resolution and (2) lack of processing power. The relatively small screen size offers more limited user interaction options than in desktop user interfaces. This problem is related to the provided screen resolution, which is also lower than in the stationary gadgets. Together this leads to less information that can be displayed and manipulated. Moreover, users of such devices are typically in motion whilst using their device. This means that they cannot devote all or any of their visual resource to interaction with applications often for safety reasons. All of these heavily affect the design and usability of user interface for presentation and interaction.

3.1 Mobile Use Model

Ease of use is central to handheld devices and their applications since mobility imposes significant cognitive and ergonomic constraints affecting devices and application usability. The usability involves many mutually dependent dimensions (Shneiderman, 1986; Holcomb & Tharp, 1991; Gould, 1988; Nielsen, 1995). Nielsen (1995) defines usability along the dimensions of learnability, efficiency, memorability, error prevention, and satisfaction. Others apply the concept of usability for systems in stationary interactions within a mobile environment to a certain extent (Ghose & Dou, 1998; Frokjaer et al., 2000; Teo et al., 2003). However, they cannot cover all factors impacting usability for mobile users who are typically engaged in tasks within dynamic environments, where the usability may be heavily influenced by the environment or context in which it is being used (Jameson, 2002; Kim et al., 2002).

Figure 3.1 shows a mobile use model that incorporates user, task, interface, context and environment. The interactions of the five elements result in distinct settings with different usability requirements and consequently different suitable interfaces. They exhibit a variety of limitations, influence each other and impact the overall usability of a mobile setting, which are discussed in the following.

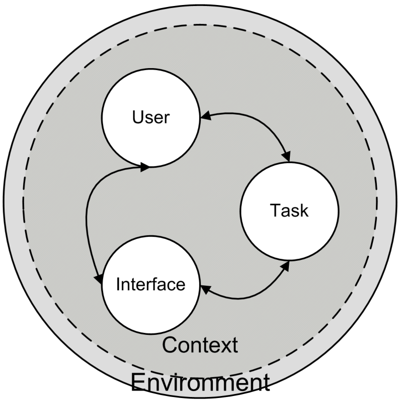


Figure 3.1: *The model of mobile use context*

The User

The user typically sets the mobile task to be completed on a particular interface within an environmental context. The expertise of this user is considered based on one experience with the task domain, system and with computerized systems in general (Kirakowsky, 1996). Expert and novice users have different characteristics in the sense of: (1) experts usually have more and specific knowledge, therefore they have more and specific demands (Rector et al., 1985), (2) experts usually have a rich set of structure within to characterize and solve new problems, (3) experts also tend to take top-down approaches to problem solving, whereas novices tend to use bottom-up approaches that lack comprehensive planning (Beaudouin-Lafon & Conversy, 1996), and (4) experts prefer the interaction on an interface to be efficient in terms of time and required actions, once these users are aware of the available commands, speed of invocation becomes a priority, while novices need to find out what commands are available and how to invoke the command.

Multi-tasking users may more easily interact with a mobile device while also interacting within their environment. In performing such activities, human users are limited among others by their memory, visual and motor skill capacities. According to Barsalou (1992), humans have a much greater capacity for recognition than for recall. The capacities and perceptions of humans' vision vary across individuals, visual stimuli and environments. Its processing can be highly dependent on user expectations. For example, if a user expects a particular image, which is presented with something that is similar (but different), the expectation may override the visual input, leading the user to incorrectly process the image (Pierce, 1955). Moreover, according to Fitts' law (Fitts & Posner, 1967), it is helpful for the user if targets are as large as possible and the distance to be moved is as small as possible. However, designing large targets on a small screen would greatly limit the number of displayed objects.

The Environment

The usability of mobile systems is likely impacted by the context in which environment is being used. In a static mobile environment, the user is engaged in applications while being stationary (for example, standing or sitting). In a dynamic mobile environment, on the other hand, the user is engaged in mobile applications while moving around (for example, walking through a mall). The level of audio and visual interferences characterizes the noise of mobile environment. Since mobile tasks are typically performed in non-(the user's) controlled environments, the user has less control over the level of this noise in the environment. Mobile applications might have to be performed under suboptimal environmental conditions, such as poor/high luminance and extreme temperatures. The conditions of the environment may change suddenly without warning. For example, users working on a train that passes through a dark tunnel. A noisy or interactive environment (for example, in a conversation) can obviously inflict additional constraints on the ability of the user to focus on the mobile task at hand. This is due to many distractions that stimulate the competition of the attention while interacting with mobile applications (Pascoe et al., 2000). The cognitive process can be altered due to time pressure, which makes users process information faster (Miller et al., 1960), more selectively (Svenson & Edland, 1987) and use less complex decision strategies (Payne & Bettman, 1988). They may put time limits for each process and simply stop when the time run out (Miller et al., 1960). Errors may occur because of the temporary overload of memory or processing capacity. Incomplete tasks or poor decisions may be resulted as a consequence of incomplete information.

The Task

A closed task can have a specific objective that is often decomposed into sub-goals. An open task has a more exploratory, vague and general objective compared to closed tasks (Carmel et al., 1992). Searching for a specific icon can be considered a closed task, whereas browsing an appropriate one on prediction results is an open task. A user may engage in a programmed task that should be processed sequentially or in emergent processing where steps are unfolded according to intermediate results. Retrieving the latest news would be considered as an access task, whereas sending an observation message is an authoring task. The task itself may pose complexity that is determined by its nature and scope as well as by the level of user involvement required. Highly-interactive tasks may be more difficult to accomplish in a mobile setting where environmental distractions and limited input mechanisms hinder interaction (see Section 2.4.3). Additionally, the availability, volume, accuracy and structure of data can greatly impact the complexity of the task (Keyes et al., 1989). Mobile users usually access to a limited amount of information compared to regular Internet users.

Task interruptions are common to mobile computing and problematic due to cognitive limitations of user attention, limiting efficiency during multitasking on a mobile device. An interruption can influence a user to alternate or switch attention (task switching) from the task to the interruption. Such multitasking places an increased burden on attention an memory. When a user attends to an interruption during a primary task, the user is serially attending to the primary and interruption task. This involves adopting a task, shifting to a different task and back to the original task (Altmann & Gray, 2000). The switching is also dependent on the type of interruption that causes a switch. Research by Cutrell et al. (2000) indicates a tendency to delay the switching until completion of a sub task as a result chunking behavior. The delay may also be related to difficulty of reconfiguring attention to the previous task, which comprises with inhibition of responses to the task, selection and activation of new intentions and schemas, and sequencing of operations in time (Gopher & Donchin, 1986). This difficulty can increase the performance time and the number of errors during task switching (Nagata & van Oostendorp, 2003).

The Interface

Handheld devices are not well suited for displaying text intensive content the same way a desktop computer is. Navigation structure is also difficult to convey. More effort is required by users to enter data/requests and errors are more likely. It is costly and time consuming if a user has to repeat an action. Additionally, there are a vast variety of handheld devices, which may differ in their screen size and quality, available input and output modalities, and the capacity of battery, memory and storage. Mobile applications that are optimized for one type of device may not work well on other devices, while mobile users may switch between platforms while carrying out a task.

Nielsen (1995); Kärkkäinen & Laarni (2002) suggested that an interface for mobile applications should offer: (1) much shorter pages, only what is required, (2) simple and explicit navigation, and (3) highly selective features, retaining only what is needed in a mobile setting. In addition, Kärkkäinen & Laarni (2002) pointed that determining the purpose of the interface based on task analysis is more suitable for mobile user interface instead of based on how it is designed for use with conventional computers. Furthermore, Vanderdonckt et al. (2001) suggested two visualization techniques to give users efficient view of the information. First, by shrinking the interaction elements, while observing usability constraints. For example, the length of an edit box can be reduced to a minimum (six characters visible at the same time with horizontal scrolling), while its height cannot be decreased below the limit of the smallest font size legible (eight pixels). Their findings include the minimum size for an icon, which is roughly 8 x 6 pixels. Many interaction elements simply cannot be shrunk to any significant extent. Therefore, second alternative is to replace the interaction element with a smaller-size alternative. For example, a Boolean checkbox typically requires less screen space than a pair of radio buttons.

A series of interaction options can be presented to the user using a menu. A table-based menu displays the options (as icons) in row and column (Figure 3.2(a)). In a tree-based menu, the options are displayed as a tree that upon selecting a top-level

Table 3.1: *Advantages and disadvantages mobile interface options*

Interface: Table-based Menu	
Advantage	***User***: (a) could support both novice and expert user, (b) relies on recognition rather than recall (Kurtenbach & Buxton, 1993), and (c) reduces the average number of key presses. ***Task***: (a) can reduce backtracking actions, (b) information structure is partially conveyed, and (c) can present more data in a tight space. ***Environment***: (a) appropriate for both quiet and noisy environments and (b) somewhat time efficient.
Disadvantage	***User***: (a) table structure is based on designer, not user's intuition and (b) quickly occupies limited screen space. ***Task***: (a) increases screen clutter and (b) may not be suited for all types of tasks, applications or information. ***Environment***: awkward within certain ambient conditions.

Interface: Tree-based Menu	
Advantage	***User***: (a) supports novice users (Shneiderman, 1986), (b) relies on recognition rather than recall (Kurtenbach & Buxton, 1993), (c) moderately reduces the average number of key presses. ***Task***: (a) reduces backtracking actions and (b) overall information structure is conveyed. ***Environment***: (a) appropriate for both quiet and noisy environments and (b) time efficient.
Disadvantage	***User***: (a) somewhat cumbersome for expert users (Shneiderman, 1986), (b) tree structure is based on designer, not user's intuition (Norman, 1991), and (c) quickly occupies limited screen space. ***Task***: (a) increases screen clutter and (b) difficult to use with a deep and/or broad tree structure. ***Environment***: awkward within certain ambient conditions.

Interface: Short-cut Menu	
Advantage	***User***: (a) supports expert users (Shneiderman, 1986), (b) highly reduces the average number of key presses, and (c) limited screen clutter. ***Task***: supports closed tasks. ***Environment***: highly time efficient.
Disadvantage	***User***: (a) does not support novice users (Shneiderman, 1986), (b) relies on recall rather than recognition (Kurtenbach & Buxton, 1993), and (c) codes are pre-determined by designer, rather than user's intuition. ***Task***: (a) does not support open tasks, (b) overall information structure is not conveyed, and (c) users may be unaware of all task options. ***Environment***: (a) awkward in environment with distractions since it requires the user's full attention and (b) awkward within certain ambient conditions.

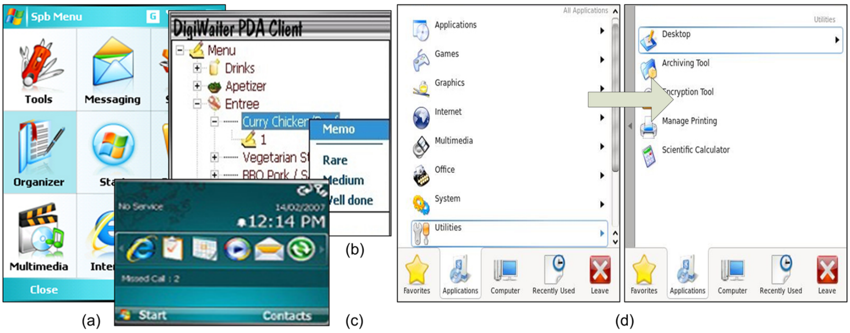
Table 3.2: *Cont'd: Advantages and disadvantages mobile interface options*

Interface: Hierarchical Menu	
Advantage	***User***: (a) supports novice users (Kurtenbach & Buxton, 1993), (b) relies on recognition rather than recall (Kurtenbach & Buxton, 1993), and (c) focuses the user on a few choices. ***Task***: (a) alerts users to the existence of task options they may be unaware of and (b) reduces screen clutter. ***Environment***: appropriate for both quiet and noisy environments.
Disdvantage	***User***: (a) cumbersome for expert users (Shneiderman, 1986), (b) increases the average number of key presses, (c) menu choices are based on designer, not user's intuition (Norman, 1991), and (d) force the user to consider the system/information in a top-down manner. ***Task***: (a) lacks visual cues for overall navigation structure (Norman, 1991) and (b) may require numerous backtracking actions. ***Environment***: (a) time consuming, (b) awkward within certain dynamic settings, and (c) awkward within certain ambient conditions.

Interface: Speech-based Menu	
Advantage	***User***: (a) could support both novice and expert users, (b) minimizes key presses, (c) hands-free, supporting multitasking, (d) natural input, and (e) not affected by visual and motor skill limitations. ***Task***: appropriate for both accessing and authoring tasks. ***Environment***: may be appropriate in various ambient conditions.
Disdvantage	***User***: (a) inappropriate for users with speech impediments or thick accents, (b) poor quality speech-synthesis may lead to user annoyance (Bousquet-Vernhettes et al., 2003), and (c) speech output is serial and provides no short-term memory aids (Schmandt, 1994). ***Task***: (a) inappropriate for tasks requiring privacy and (b) inappropriate for tasks containing ambiguous words ***Environment***: may be influenced or degraded by the environment.

node within a tree-structure, it is expanded to show sub-topics available within this option (Figure 3.2(b)). Users can jump to sibling and parent content with a single action. A set of options in a short-cut menu is directly associated with certain applications, functions or information (Figure 3.2(c)). The user usually chooses this set beforehand. In a hierarchical menu, when one of the options is selected a series of sub options is shown (Figure 3.2(d)). Users can backtrack or move forward to see available options. Using a speech-based menu, a selection of presented options can be inputted verbally. Table 3.1 - 3.2 presents the advantages and disadvantages of these interface types within the context of the ubiquitous usability model (Hassanein & Head, 2003).

Novice users who require guidance through a navigation system may often use a hierarchical menu, while experts who wish to reduce the average number of key presses and task time may use short-cut menu. A table-based and a speech-based



(a) (b) (c) (d)

Figure 3.2: *Interface options for mobile applications: (a) table-based, (b) tree-based, (c) shortcut menu, and (d) hierarchical menu*

interface may support both novice and expert users. The latter offers handsfree and supports multi-tasking for both novice and expert users. It could be applied to any of the other interface modes. The preferences for the interface options may vary according to the user, task and environment. Such settings do not remain constant. Moreover, a user may progress from novice to expert. This user may engage in various types of tasks within a dynamic environment.

The Context

Context is any information that can be used to characterize the situation of an entity (Abowd et al., 1999). An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application itself (Perlovsky, 1999). Context encompass the physical environment, the user, the task, and the interaction between the three on the interface (Tarasewich, 2003). From the perspective of a mobile interface, context can be used to increase the performance of user input (less input, faster, and less error) (Uther, 2002). The context can support adaptive navigation and compact yet still relevant information display. In facilitating information search and knowledge discovery, for example, such an interface can assist the user in acquiring the most salient information in any particular context, in the best form and the most appropriate time. Without context, it would be proved that the user is working in an inefficient way during "hunting" for the correct information. How well this context is understood can impact the way a user interface can improve the quality of service or formulate the strategy of action selections or information presentation.

3.2 Adaptive User Interface Model

Proposed by Rouse (1988) in the psychological literature, the concept of an adaptive user interface (AUI) specifically tailored for the HCI systems emerged as an intelligent user interface. An AUI can be described as a set of displays and controls, user and an underlying software system that is capable to change over time in response to how it is used to improve the quality of the interaction with its user (McTear, 2000). It can look at the current task, understand it, recognize the user's intention and automatically take over the task completely or partially, allowing the user to focus on other more important activities (Alvarez-Cortes et al., 2007).

Research in AUI has offered mobile technology opportunities by removing temporal and spatial constraints as well as providing interaction and knowledge content adaptabilities (Billsus et al., 2002; Rothrock et al., 2002). Interaction adaptability is required for an execution and control task (manual or supervisory control) while knowledge content adaptability is needed for knowledge acquisition and refinement tasks (for example, information search, knowledge discovery and collaboration). In the case of knowledge content adaptability, the requirements of interface design focus on the elicitation of information from the user and the refinement of information based on personalization and patterns of interaction. The information overflow associated with finding information in complex systems or large databases can be reduced through the use of AUIs. Irrelevant information can be filtered out, therefore reducing the user's cognitive load. These are enabled via assumptions made about user behavior after analyzing patterns discovered through implicit data gathering (Alvarez-Cortes et al., 2007). Tasks are made easier by modifying the communication style, content and form of information that is displayed.

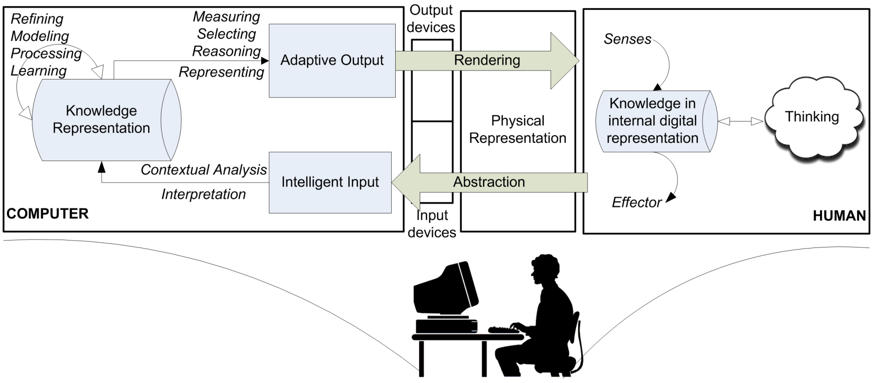


Figure 3.3: *A model of human-computer interaction on an adaptive user interface*

Compared to the traditional HCI (Figure 1.1), an AUI in Figure 3.3 is able to provide prediction to a user based on its experience with this user. It considers the style of communicating, which reflects on interaction choices and message compositions in inputs, of each individual user. Such an interface can offer faster input with high accuracy. The information can be adapted and presented to suit the user's activities and context of use. At some level of detail, a combination of measured and modeled user behavior is analyzed and refined to provide an interaction that fits the behavior as best as it can. The increased knowledge provides the interface the ability to interact with more intelligence, more flexible and with the ability as provided by direct measurement, in-line models of the human or both. Combined with the abstraction results, the analysis results are stored in the same internal representation. In the opposite of the transformation process, by assessing the system's belief about state of the user, the task and the context, the output side selects appropriate contents, produces context-aware and user-tailored output representation and renders it into a raw description of the representation that is appropriate for the available modalities.

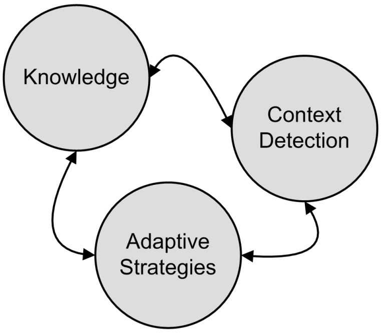


Figure 3.4: *The model of an adaptive user interface*

Generally, the model of an AUI is as depicted in Figure 3.4 that incorporates context, model and adaptation strategies. This model can be applied in any adaptive component of an application or any part of an interface. An AUI differs from a non-AUI in that is more knowledgeable of the individual characteristics of the user, of the implications of interactivity between the user and the interface, and of environment and interface effects on the user. This typically is achieved by using dedicated modules responsible for acquiring and recognizing the user's context, maintaining the knowledge content and reasoning towards suitable interface adaptations (McTear, 2000), as explained in the following.

Context Detection

Location, identity, environment, activity and emotion are the primary context variables for characterizing the situation of a user (Pantic et al., 2006) (see Section 2.4.2). The context not only answers the questions of who, what, when, where, how, and why, but also acts as indices into other sources of contextual information. With an entity's location, what other objects or people are near the entity and what activity is occurring near the entity can be determined. The context of information about user's emotional state can be used to characterize the situation of the user or an environment. The context of activity may give information on user intention and need. In an adapt-able system, the user must deliberately choose the adaptation by explicitly specifying their preferences, typically before the interaction begins. Such a system allows the user to modify certain aspects of the interactive behavior (McTear, 2000). Whereas, in a dynamic AUI, the user's behavior and environment is monitored. Dedicated input modules supply an interpretation module, which constructs the system's beliefs about the user's context, needs and intention.

Adaptation Strategies

Selecting suitable adaptation can decrease both the time and expertise of the user, which is required to interact with an application. Most relevant adaptation strategies that work most efficiently for different context and user profile are selected and executed by various adaptation processes. These processes perform computation and reasoning on recognized user context and presented knowledge content. They are usually independent (automatic) of the control of the application. Different adaptation procedures may also been applied in different parts of the interface. The procedures usually use knowledge of belief in the state of the environment and have facilities to discover from patterns of behavior from the user. For example, novice users can be overwhelmed by the volume of available data inherent in complex operational environments. Their capability to attend, process, and integrate information efficiently will be decreased resulting in information overload and negatively impacting their situation awareness (Endsley, 1995). Filtering information is necessary. In contrast, experienced users may need all information to assess and interpret the current situation. For these type of users, cues in the provided information can guide their decision making process.

Knowledge

Developing the system's knowledge involves the use and articulation of reusable models and knowledge repositories encapsulating the wide variety of details pertaining to the user interaction. Whereas, maintaining the knowledge content involves storing and learning traces of the user's interaction. The knowledge usually contain models of belief in the state of the task and environment and have facilities to discover from patterns of behavior from user(s). The knowledge can contain a device model (that incorporates different platforms and different technology capabilities), a user model (that is specific to different roles, goals, preferences), a task model (that is for different conditions and states), and a world model (that is for different particular domain and context of use).

3.3 Multimodal Interface

In a communication act, whether it is between humans or between computer system and a user, both modality and mode come into play. The modality defines the type of data exchanged whereas the mode determines the context in which the data is interpreted. Characteristics of multimodal communication are the multiplicity modes and multimodality, which is supported by multiple coordinated activities at various cognitive levels. As a result the communication becomes highly usable and robust. Failure of one mode is recovered by another mode and a message in one mode can be explained by another mode (Nagao & Takeuchi, 1994). A user of a multimodal computerized system can combine multiple modalities to transfer a single message, which has been shown to decrease error rates (Johnston & Bangalore, 2000; Oviatt & van Gent, 1996).

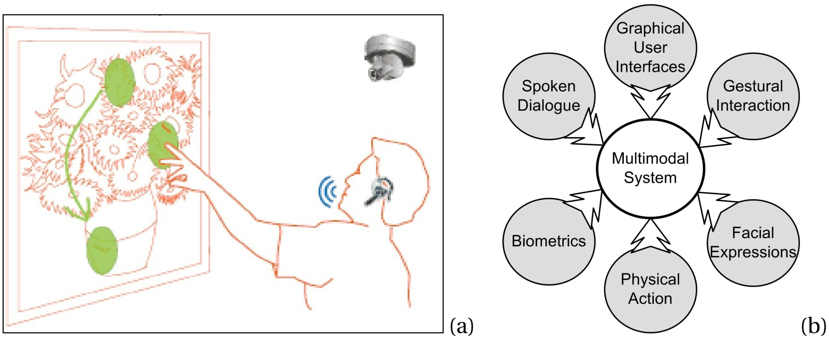


Figure 3.5: *(a) A multimodal application: Museum Recorder (Amsbary, 2007) and (b) merging various dialogue and interaction paradigms into multimodal systems (Wahlster, 2006)*

Compared with traditional command line interfaces, multimodal interfaces are able to support rich expressiveness using familiar communication modalities. Figure 3.5(a) shows an example of interfaces that are designed to support simultaneous use of input/output modes. These interfaces can be designed to permit switching among modes to take advantage of the modality best suited for the user's capabilities, a task and environment, using one to enhance and complement another. A multimodal interface is built by integration of various sensors and modalities one to enhance and complement another, as depicted Figure 3.5(b). For example, the speech and lip reading interface has demonstrated to substantially reduce speech recognition errors and also to stabilize system reliability in noisy field environments (Wojdel, 2003).

Multimodal interfaces have implications for accessibility toward large individual differences present in the population. A well-designed multimodal interface can be used by people with a variety of impairments (Oviatt, 2001). Visually impaired users rely on the voice modality with some keypad input. Hearing-impaired users rely on the visual modality. Such interfaces are also inherently flexible and ideal for accommodating both the changing demands encountered during mobile use (Oviatt, 2000) and task switching (Oviatt et al., 2004). They offer various ways of user interaction in a more natural fashion with provided services. These interfaces are a clear requirement for universal access.

3.3.1 Multimodal Systems: Early History

Systems that use more than one channel/modality to communicate information are called either multimedia or multimodal systems. The difference is that multimodal systems use a higher level of abstraction from which they generate output and to which they transform the user input (Coutaz et al., 1993). These systems comprise additional components for processing information from multiple input modalities and the output of such a system is usually richer as a screen is used to present multimedia in synchronized manner.

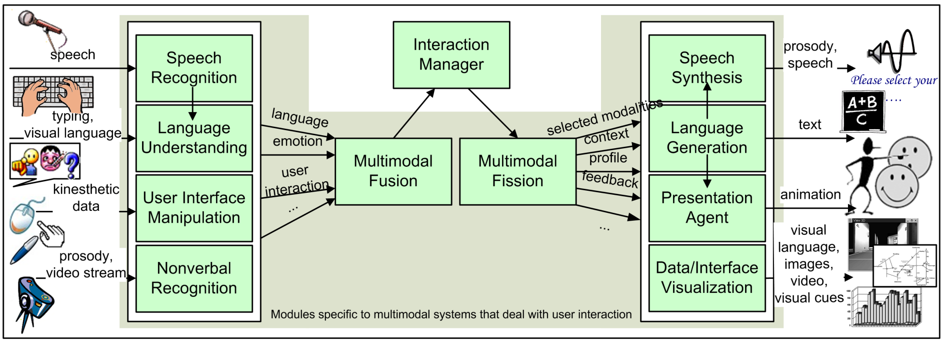


Figure 3.6: *Pipeline architecture of a multimodal system*

The multimodal input are combined in the fusion component; the user system interaction flow is controlled by an interaction manager; and multimodal output is controlled by a fission component. This classification is the most traditional classification, but it cannot be considered as the ultimate one. Due to practical reasons, most multimodal systems very often combine the response generation as a part of an interaction manager. These systems also often incorporate other additional modules, for example a module that chooses most relevant modality considering the context or a module that processes external knowledge. As far as the organization of a multimodal system components is concerned, the pipeline architecture is the most simple one. Modules are simple connected linearly into a sequence in the order they process the input, as shown in Figure 3.6. In the following, several key multimodal systems are discussed by focusing on their approaches that pertain to user interface domain.

SmartKom

SmartKom combines speech, gesture and facial expressions for input and output (Figure 3.7) (Wahlster, 2006). It works in multiple domains, such as a TV program guide and a mobile travel companion, using "Smartakus", a conversation agent. This system uses M3L (Multimodal MarkUp Language) to cover all data interfaces within the system. The presentation planning component decomposes the presentation goal recursively into primitive presentation tasks using some predefined presentation strategies with the discourse context, the user model, and ambient conditions. This component also specifies presentation goals for the language generator, the display and the character animation. The language generation follows a template-oriented approach based on Tree Adjoining Grammar (Joshi & Schabes, 1997). The output consists of text annotated with conceptual structures for a concept-to-speech synthesizer. The animation generator selects appropriate elements from a catalogue of basic behavioral patterns to synthesize actions of the Smartakus. All planned deictic gestures of the Smartakus are synchronized with the graphical display of the corresponding media objects. Its facial animation is also synchronized with the planned speech output. The display manager generates an M3L representation of the current screen content by adding the corresponding linguistic and visual objects to the discourse representation.

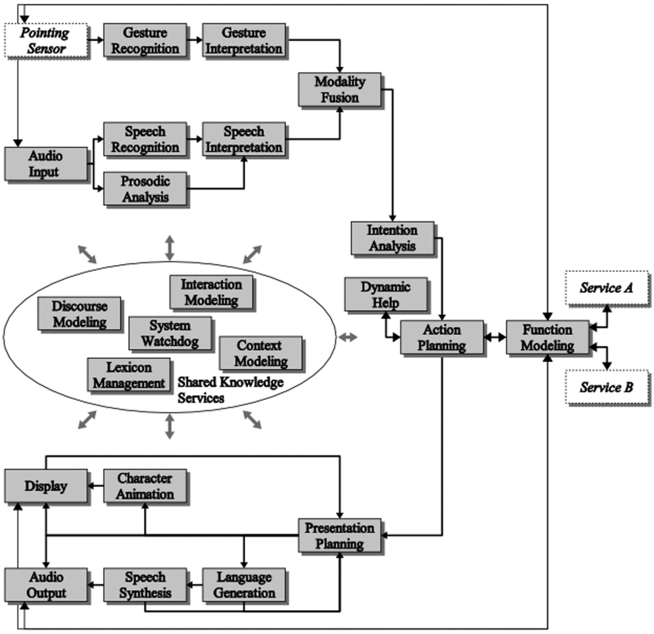


Figure 3.7: *SmartKom Architecture*

COMIC

The project demonstrator of the COMIC project adds a dialogue interface to a CAD-like application used to help clients redesign their bathrooms. The input includes speech, handwriting, and pen gestures (Figure 3.8). The output combines synthesized speech, a talking head, and control of the underlying application. Knowledge of the application is stored in an ontology represented in RDF/OWL. To create the textual content of a description, the output module proceeds as follows (Foster et al., 2005). First, it gathers all properties of the specified design from the ontology. Next, the module selects the properties to include in the description, using information from the dialogue history and the user model, along with any properties specifically requested by the dialogue manager. This module then creates a structure for the selected properties and creates logical forms using factored language models over words and multimodal co-articulations to select the highest-scoring realization licensed by the grammar that satisfies the specification for the output module. It also incorporates facial behavior specifications (expressions and gaze shifts) and deictic gestures at objects on the application screen using a simulated pointer. The output module controls the system output using the timing information returned by a speech synthesizer to create a full schedule for the turn to be generated.

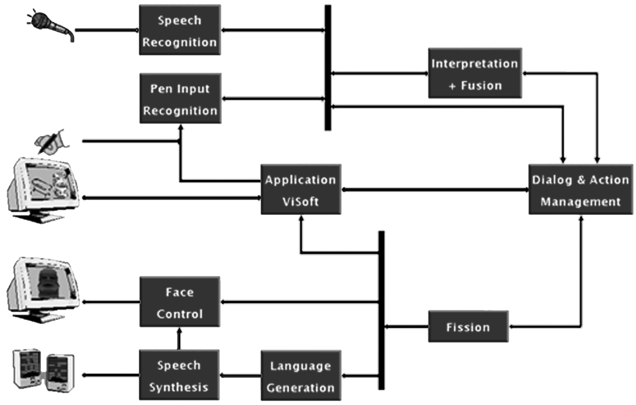


Figure 3.8: *Comic system architecture*

A Gesture and Speech-Driven Crisis Management System

Sharma et al. (2003) proposed a framework that employs the processing of natural gesture and speech commands elicited by multiple users to manage dynamic emergency scenarios on a large display. Their project demonstrator, XISM, allows a user (an operator of an emergency center) to dispatch emergency vehicles to crisis locations in a virtual city indicated by animated symbols and accompanying audible signals. The response generation in the design concept of XISM has four components (Figure 3.9): plan reasoning, information control, (mixed-initiative) dialogue control and response content assembly. With direct access to three knowledge sources: user knowledge, task knowledge and world knowledge, the plan-reasoning serves two main purposes: (1) to establish the system's intention and belief; and (2) to elaborate the plan on the course of actions for the task in focus. The information control interacts with the external information systems on one side and communicates with the dialogue control and response content assembler on the other side. A dialogue can be user-led, system-led or mixed initiative. The last component is responsible for organizing the response contents into a schema that is usable by the response presentation module. The prototype covers different computing platforms, communication devices and network connectivity that vary from location to location. A device detection engine differentiates and determines the type of user device. The data appropriately for each device is processed in the multimedia engine, so that the size and the type of the data are always compatible. All communication data is composed in XML.

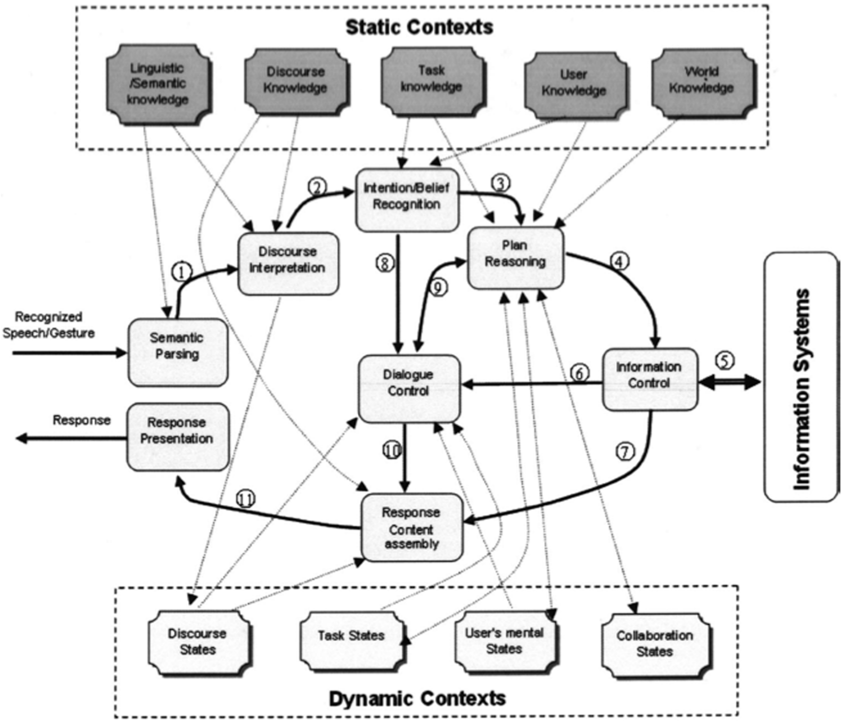


Figure 3.9: *The design concept of the dialogue management in XISM*

MATCH - Multimodal Access To City Help

MATCH provides input using speech, handwriting, touch or composite commands combining multiple modes, such as a printer (Johnston et al., 2001). The test-bed interface provides restaurant and subway information of a city using a lifelike graphical talking head. The underlying architecture of MATCH consists of components, which communicate using XML messages sent over sockets through a facilitator - MCUBE (Figure 3.10). Once a command is recognized, the dialogue manager passed it to a preprocessing component - MMGEN, which builds a multimodal score indicating a coordinated sequence of graphical actions and TTS prompts. The interpretation of input includes the dialogue history, focus space, models of user and system beliefs.

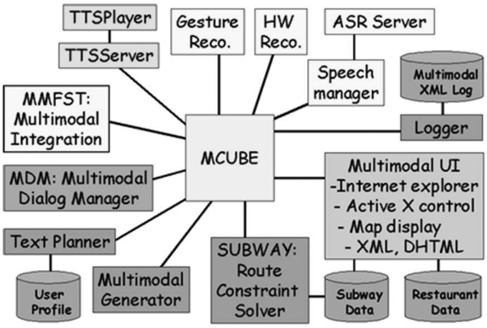

TTSPlayer
TTSServer
Gesture Reco.
HW Reco.
ASR Server
Speech manager
Multimodal XML Log
MMFST: Multimodal Integration
Logger
MCUBE
MDM: Multimodal Dialog Manager
Multimodal UI -Internet explorer - Active X control - Map display - XML, DHTML
Text Planner
SUBWAY: Route Constraint Solver
User Profile
Multimodal Generator
Subway Data
Restaurant Data

Figure 3.10: *Match system architecture*

A selection process determines the system's next move using ontology, which contains information about different types of actions. A template-based generation processes simple responses and updates the system's model of the user's intentions after generation. The text planner is used for more complex generation, for example comparing two restaurants based on user preferences. The Multimodal UI passes prompts to a visual text-to-speech component to synchronize TTS with the lip movements. When the UI receives a multimodal score, it builds a stack of graphical actions. It then sends the prompts to be rendered by the TTS server. After prompts are realized, the Multimodal UI receives notifications and presents coordinated graphical actions.

MACK - Media lab Autonomous Conversational Kiosk

Figure 3.11 shows MACK, an embodied agent who can answer question and give directions in mixed reality and kiosk format (Cassell et al., 2002). It uses multimodal output with speech synthesis, gesture and an LCD projector output directed at the physical map. BEAT is responsible for the generation of appropriate speech with intonation, hand gesture, and head and eye movements (Cassell et al., 2001). MACK uses a template-based sentence generator, which allows abstract response categories from the Reaction Module to be translated into text and sent to the BEAT for the generation appropriate nonverbal behavior. The mapping from the text to facial, intonational and body gestures is contained in a set of rules. The annotation allows generation, synchronization and scheduling of multiple nonverbal communicative behaviors with speech. The BEAT provides two ways to achieve the synchronization. The first is to obtain estimates of word and phoneme timings and construct an animation schedule prior to execution. The second is to assume the availability of real-time events from a TTS engine generated while the TTS is actually producing audio.

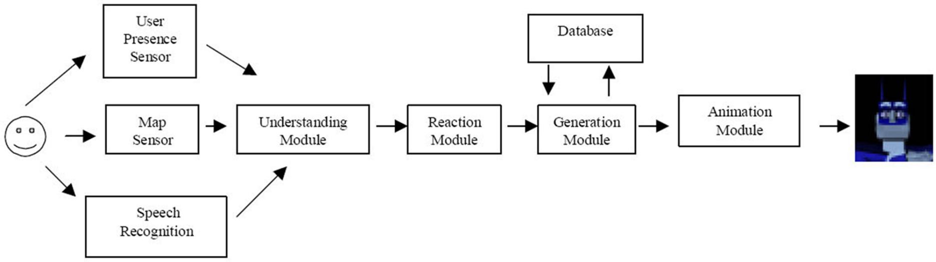


Figure 3.11: *MACK system architecture*

AdApt

AdApt project provides speech and audio/visual information in real-estate domain using a static 3D wireframe of an animated face (Gustafson et al., 2000). Additional input can be done by clicking icons on an interactive map. All components in the system communicate via a broker using TCP/IP sockets. The project exploited XML encoded messages for the communication between modules. The dialogue manager (DM) sends a dialogue act to I/O manager (Figure 3.12). The I/O manager was implemented to facilitate turn-taking in the dialogue. It also handles the timing in the system for coordinating verbal and graphical output. The I/O manager decomposes the system responses from the DM into commands that are sent to the different output modules. Each message consists of two parts: a feedback and a constraint history. The

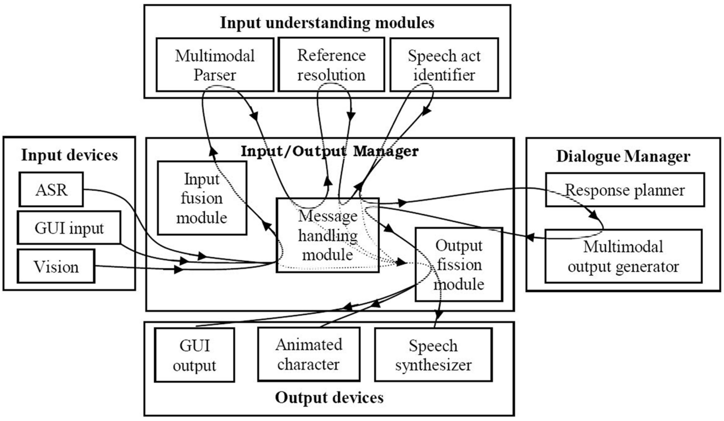


Figure 3.12: *The architecture of AdApt*

feedback part consists of three subparts: (1) a verbal paraphrase of the constraints in the user's latest utterance, (2) a list of names on icons representing these constraints, and (3) a list of apartments found using the current constraint. The I/O manager assesses the recognition confidence score to decide which kind of feedback strategy to use. The GUI manager provides a common frame for the animated talking head, the map handler and the icon handler.

3.3.2 Multimodal Representations for a Multimodal System

A model for HCI consists of a number of components that, in most of the cases, were developed independently. Those components usually are designed around the task that the components are targeting. Decisions on the platform, the programming language to use, the interface and the communication infrastructure to other components are made mainly with the targeted problem in focus. Integrating all components in such a framework is a challenging task, due to: (a) the components may provide different interfaces; (b) each component may deliver or expect different format of data; and (c) the components may have variety level of robustness and reliability. To tackle these problems the specification of a language that represents both the form and the content of linguistic resources is important.

In recent times a growing number of researchers focus on the development of representation for the communicative acts between autonomous components in a multimodal system. They approach this topic from different viewpoints. Some use typed input with synchronized nonverbal behaviors, others use manually inserted or automatically determined during a linguistic and contextual analysis of the text. These approaches identify speech as the dominant modality responsible for time structure of the utterance. Synchrony is achieved by starting the animation of nonverbal behaviors, such as facial expressions, simultaneously with the correlated verbal phrase. The nonverbal behaviors are referred to by unique identifiers and are drawn from a behavior database. Another approach is to build nonverbal ontology that allows to create nonverbal behaviors from atomic elements and to adapt their structure in the synchronization process. For the sake of flexibility and adaptability, most of the approaches are an XML-based language. Some examples of knowledge representations are discussed in briefly in the following.

M3L - Multimodal Markup Language

M3L is specified for the SmartKom project. It covers 40 schema specifications. As a foundation, an ontology is build coded in OIL (Ontology Interface Language). A tool called OIL2XSD (Gurevych et al., 2003) transforms the ontology into an M3L. Its top level of ontological categorization divides all concepts into the `Type` class and the `Role` class (Figure 3.13). The `Type` class subsumes everything that is independent of how they are applied and always has the same ontological status independent of the particular context. The `Role` class subsumes anything that can take on a different role in a specific situation, event or process. The `Role` class is divided into the `Event` class and the `AbstractEvent` class. The `Event` class is any entity that exists in space and time, such as any physical object or process. The `AbstractEvent`'s subclasses, either an `AbstractObject` or an `AbstractProcess`, do not exist relative to any conceptualization of space or time, for example abstract number, abstract representational object, and set. In particular, M3L represents all information about segmentation, synchronization and confidences in processing results. The basic data flow of user input to system output adds further processing results so that the representational structure will be refined, step by step. SmartKom uses unification, an operation called overlay (Alexandersson et al., 2006), of typed feature structures encoded in M3L for discourse processing.

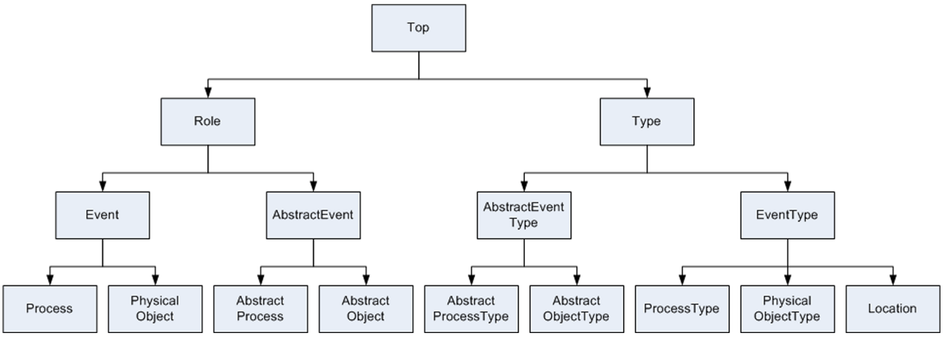
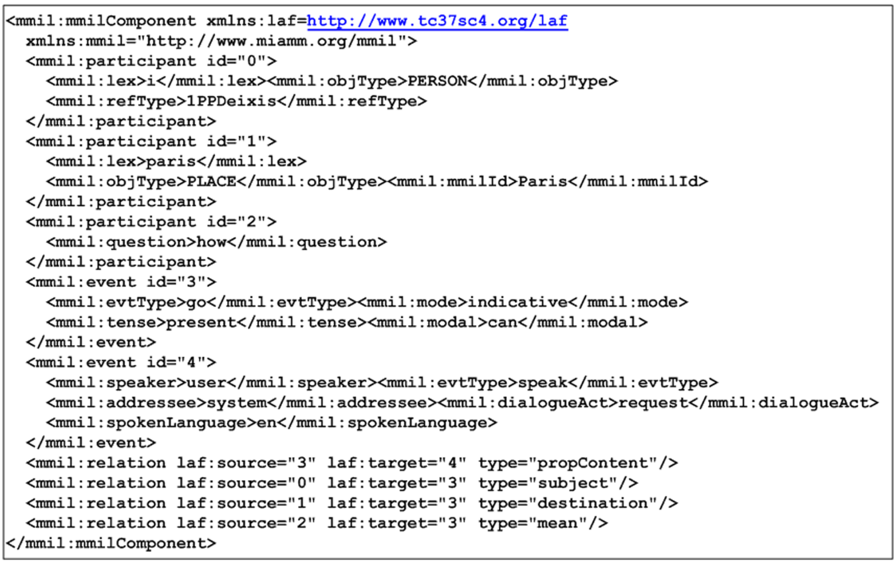


Figure 3.13: *Upper Ontology used in the SmartKom project*

MMIL - MultiModal Interface Language

The specification of MMIL represents semantic content in a multimodal context to capture: (1) linguistic, gestural and graphical events, and (2) dialogue acts and their contents. It aims at providing a meta-model for semantic representation free from any modality constraint. The MMIL meta-model abstracts different level of dialogue information (phone, word, phrase, and utterance) by means of a flat ontology, which identifies shared concepts and constraints. The definition layer of the ontology includes two kinds of entities, such as `event` (an object that is associated to the temporal level) and `participant` (a static entity that is acting upon or being affected by the events). Dependencies between entities are represented as typed relation linking structural nodes (Figure 3.14). The relation between entities are implicitly represented by qualifying descriptors defining anchors among entities. The other information, such as morpho-syntactic, domain, annotation description, are formed as data categories expressed in an RDF format. MMIL distinguish three different speech acts: saying, telling, and asking (*wh*-questions).



```
<mmil:mmilComponent xmlns:laf=http://www.tc37sc4.org/laf
  xmlns:mmil="http://www.miamm.org/mmil">
  <mmil:participant id="0">
    <mmil:lex>i</mmil:lex><mmil:objType>PERSON</mmil:objType>
    <mmil:refType>1PPDeixis</mmil:refType>
  </mmil:participant>
  <mmil:participant id="1">
    <mmil:lex>paris</mmil:lex>
    <mmil:objType>PLACE</mmil:objType><mmil:mmilId>Paris</mmil:mmilId>
  </mmil:participant>
  <mmil:participant id="2">
    <mmil:question>how</mmil:question>
  </mmil:participant>
  <mmil:event id="3">
    <mmil:evtType>go</mmil:evtType><mmil:mode>indicative</mmil:mode>
    <mmil:tense>present</mmil:tense><mmil:modal>can</mmil:modal>
  </mmil:event>
  <mmil:event id="4">
    <mmil:speaker>user</mmil:speaker><mmil:evtType>speak</mmil:evtType>
    <mmil:addressee>system</mmil:addressee><mmil:dialogueAct>request</mmil:dialogueAct>
    <mmil:spokenLanguage>en</mmil:spokenLanguage>
  </mmil:event>
  <mmil:relation laf:source="3" laf:target="4" type="propContent"/>
  <mmil:relation laf:source="0" laf:target="3" type="subject"/>
  <mmil:relation laf:source="1" laf:target="3" type="destination"/>
  <mmil:relation laf:source="2" laf:target="3" type="mean"/>
</mmil:mmilComponent>
```

Figure 3.14: *MMIL file for a question*

MURML - Multimodal Utterance Representation Markup Language

MURML bridges between the planning and the animation tasks in the production of multimodal utterances of an anthropomorphic agent - Max (Kranstedt et al., 2002). It provides the description of gestural behaviors and merges the representations from interpretation and generation. The gesture animation process builds on a hierarchical model of planning and controlling the upper-limb movement of Max based on Kendon (1980) by defining units of gestural movements. Speech is produced in several locutions-over intonational phrases, which are separated by significant pauses and display a meaningful pitch contour with exactly one pitch accent being most prominent (the nucleus). Figure 3.15 shows a parse tree of an utterance. The verbal part of a complex multimodal utterance that comprises multiple gestures must be divided

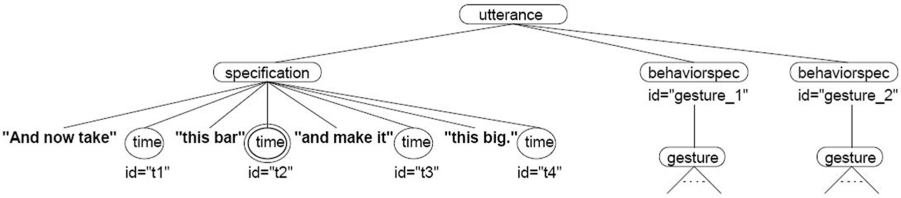


Figure 3.15: *Tree structure of utterance in MURML*

into chunks by annotating the corresponding time tags. In the subsequent definition of the nonverbal behavior, correspondence between speech and gesture is expressed by specifying the affiliate's onset and end. Gestures are stated by specifying a required communicative function sufficient for the agent to choose an appropriate behavior from a gesture lexicon. The desired gestured are described explicitly in terms of its spatio-temporal features, such as the location and orientation of hand and wrist.

BEAT

The BEAT system operates by reading in XML-tagged text representing the text of a character's script and converting it into a parse tree (Cassell et al., 2001). The root of the tree is the UTTERANCE, which is operationalized as an entire paragraph of input (Figure 3.16). The utterance is broken up into CLAUSEs to represent a proposition. Clauses are divided into two smaller units of information structure, a THEME (a coherence link with a preceding clause) and a RHEME (a part that contribute some new information to the discussion). The next to smallest unit is the word phrase: either an ACTION (verb phrase) or an OBJECT (noun phrase). The BEAT exploits WordNet (Fellbaum, 1998) to find matching sense of a word, if the exact match cannot be found in the ontology. The behavior generation module suggests n-best behaviors and uses a filter to trim them down to a set appropriate for a particular character. The behavior scheduling module converts its input tree into a set of instruction, which can be executed by an animation system.

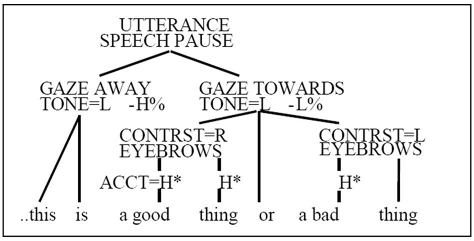
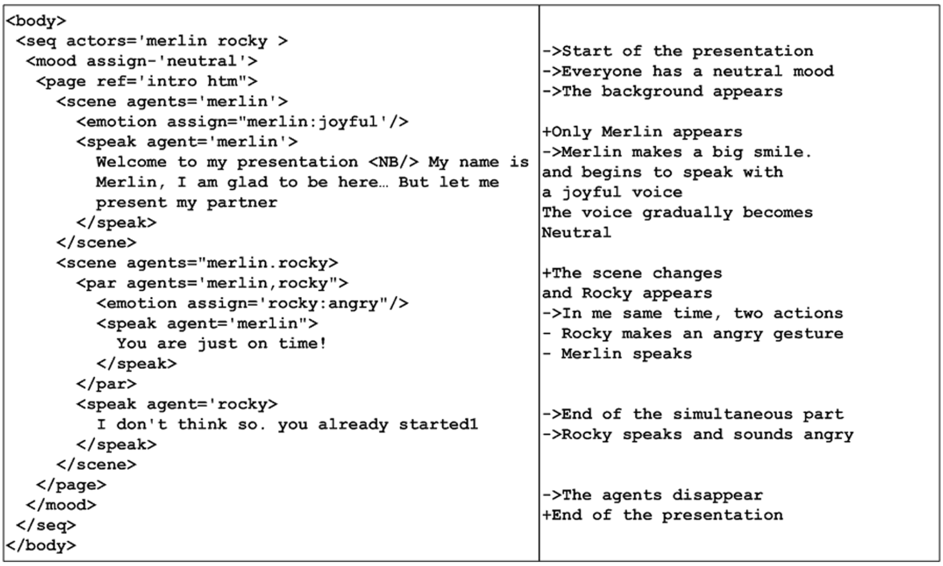


Figure 3.16: *BEAT's text and nonverbal behavior tree*

MPML - Multimodal Presentation Markup Language

MPML is designed for multimodal presentation using interactive lifelike agents (Tsutsui et al., 2000). It aims at providing a means to write presentations easily. It illustrates a procedural description of Preprocessing. The organization of the presentation in MPML is accomplished by a graph-like structure, indicated by <SCENE> tag (represents a scene of a script), <SEQ> tag (sequential actions), <PAR> tag (simultaneous actions) and <PAGE> tag (the presentation page) (see Figure 3.17). MPML defines action tags, such as <THINK>, <SPEAK>, <PLAY> and <MOVE>. User system dialogue of this presentation is developed based on AIML (Artificial Intelligence Markup Language) (Wallace, 2003). By a reasoning engine, the system can track the conversation, monitoring both user input and responses and intervening based on changes in the conversation conditions.



```
<body>
 <seq actors='merlin rocky >
  <mood assign-'neutral'>
   <page ref='intro htm">
     <scene agents='merlin'>
       <emotion assign="merlin:joyful'/>
       <speak agent='merlin'>
         Welcome to my presentation <NB/> My name is
         Merlin, I am glad to be here… But let me
         present my partner
       </speak>
     </scene>
     <scene agents="merlin.rocky>
       <par agents='merlin,rocky">
         <emotion assign='rocky:angry"/>
         <speak agent='merlin">
           You are just on time!
         </speak>
       </par>
       <speak agent='rocky>
         I don't think so. you already started1
       </speak>
     </scene>
   </page>
  </mood>
 </seq>
</body>
```

```
->Start of the presentation
->Everyone has a neutral mood
->The background appears

+Only Merlin appears
->Merlin makes a big smile.
and begins to speak with
a joyful voice
The voice gradually becomes
Neutral

+The scene changes
and Rocky appears
->In me same time, two actions
- Rocky makes an angry gesture
- Merlin speaks


->End of the simultaneous part
->Rocky speaks and sounds angry


->The agents disappear
+End of the presentation
```

Figure 3.17: *MPML script illustrates the effect of emotional tags*

3.4 Overview: An Adaptive User Interface Framework

The research reported here proposes a framework that allows the rapid construction and evaluation of adaptive HHI interfaces. The development of the framework aims at modules integration that is independent of the availability of modalities. The current development focuses on a framework for constructing a HHI interface that is able to support communication between different users via handheld devices, such as sending each other messages. To support the adaptability and mobile use applicability of the framework, three keys aspects of an intelligent user interface are offered: *multimodality*, *context-awareness* and *affective*. The first two aspects are described in the following. The affective aspect is covered in these descriptions.

3.4.1 Multimodality of Human Handheld-Device Interaction

Communication technology is information technology that deals with the most complex information medium in our world: human language. Human language occurs in spoken and written form. Whereas speech is the oldest and most natural mode of language communication, complex information and most of human knowledge is maintained and transmitted in written texts. Of course, this is not completely the case, since people write mails and SMS or join text-based chat rooms. As human combines speech with gesture and facial expressions, digital texts are combined with pictures. In communication, language is mixed with other modes of communication and other information media. This allows communication technologies to facilitate processing and generating multimodal communication and multimedia artifacts. Along with the steady stream of audio and video that surrounds people in everyday life, it could be said that information today is still mostly stored using a combination of text and static images, such as picture, diagrams, maps and the like. The main reason is that the tradition of paper-based writing and reading still remains. Along with this consideration, the research reported in this thesis focuses on investigating methods and input recognition modules of (natural language) text and visual language-based messages.

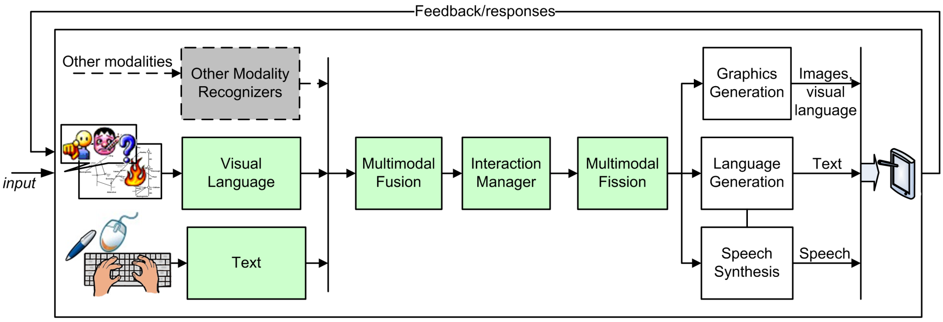


Figure 3.18: *The architecture of an Adaptive-Multimodal User Interface Framework*

Figure 3.18 shows the architecture of the framework. The modules in the framework deal with HHI, the interpretation of inputs, knowledge structure and management, the generation of appropriate responses, and the presentation of these messages. Every module is able to work in real-time and in isolation, but also as part of the framework. The output of the framework combines text, synthesized speech, visual language and the control of the underlying user interface.

The design of the framework allows input modalities to be added in an ad hoc, as long as the output of the additional input recognizers follows the notion as described in the framework's knowledge representation. The multimodal input from each user is interpreted and combined in the fusion component. This process includes the analysis of the user's emotional state. The interaction manager (IM) generates appropriate responses and the fission component displays these using synchronized modalities depending on the particular user's context. The latest includes allocating and coordinating information across media, such as typed or spoken language, visual language, and graphics.

The architecture of the framework covers three different fields of research: the field of user input recognition, of message interpretation (context-aware computation), and of (output) information generation and presentation (Figure 3.19). To be able to form cognitive awareness of the (user's) world and to support consistent communication data inter-components, the knowledge representation about the user, the task and the world is defined as ontology. The cognitive awareness of the world itself is assumed to be represented in the human mind as many scripts. A script represents the chain of events that identify a possible scenario. In a script, a scenario is distinguished by some conditions. These conditions are defined consistently using concepts and their properties in the ontology. Both ontology and scripts are utilized in all processes of input recognition, of message interpretation, and of information generation and presentation.

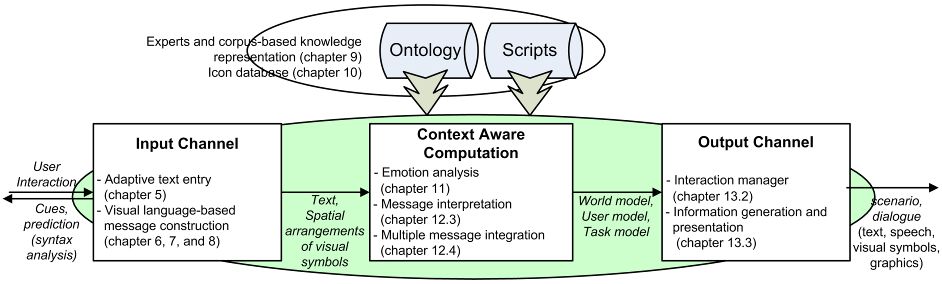


Figure 3.19: *Research coverage (with chapter no.)*

The framework offers a rapid development environment to create HHI interfaces that perform specific tasks within a certain domain. Depending on its context of use, the architecture of the framework may have different structure. Currently, a demonstrator interface for reporting observations in a crisis event is developed. It is able to collect up to date observations, interpret them automatically and form a global view about the reported events. Figure 3.20 shows the mobile use of context of this interface based on the general model in Figure 3.1. The system is designed to support various roles within the field of crisis management, including professionals and civilians in the field and the control room. The input from various input recognitions are processed into a coherent and context dependent interpretation representation. The user interaction on the developed interface involves the employment of ontology and scripts in the environment of multiple users, multiple devices and multiple modalities.

3.4.2 Context-Awareness of Human Handheld-Device Interaction

In dynamic situations, the states of the user, the task and the environment can change from moment to moment, such as those found in a mobile user interaction. A real-

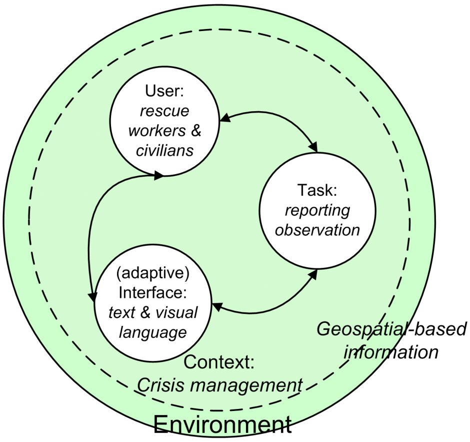


Figure 3.20: *The model of mobile use context in the field of crisis management*

time rapidly customizable interface between the user and technology is required to continuously maintain the best match between these entities for maximizing the interaction of the actor and the technology. An AUI is able to adapt to the user's state in real-time and provide the user with the right information at the right time. Although interaction controls should be still in human user hands, nevertheless, accurate and easy access of information can support better decision, planning and reasoning and situation awareness of the users. The performance enhancement provided by the AUI can lead to more effective and efficient interaction, as well as reduce training requirements by enhancing the novice user's performance ability.

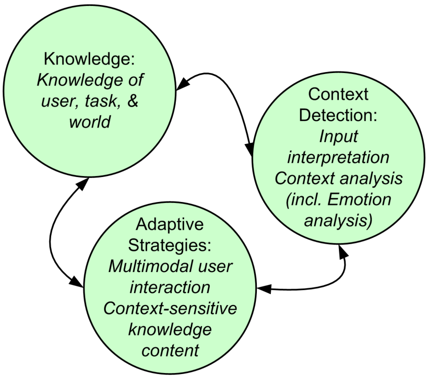

Knowledge:
Knowledge of user, task, & world
Context Detection:
Input interpretation
Context analysis
(incl. Emotion analysis)
Adaptive Strategies:
Multimodal user interaction
Context-sensitive knowledge content

Figure 3.21: *The model of an adaptive user interface*

The model of an AUI of the proposed framework is depicted in Figure 3.21, which is based on the general AUI model in Figure 3.4. The ability to adapt its behavior and interaction with individual users is achieved by using dedicated modules responsible for: (a) personalizing the library of input prediction, (b) interpreting input and analyzing the user's context variables including the user's emotional state, (c) maintaining the internal knowledge, and (d) reasoning and applying of suitable interface adaptation strategies. Two types of adaptation strategies are offered by the proposed framework, are: (1) adaptability of input and output (user interaction) and (2) adaptability of knowledge content.

Adaptive User Interaction

Using a handheld device, a user can communicate with others. The user can create messages on the provided interface using available modalities. Figure 3.22 shows the applicability of the proposed framework into an AUI in mobile domain, specifically for interaction on handheld devices.

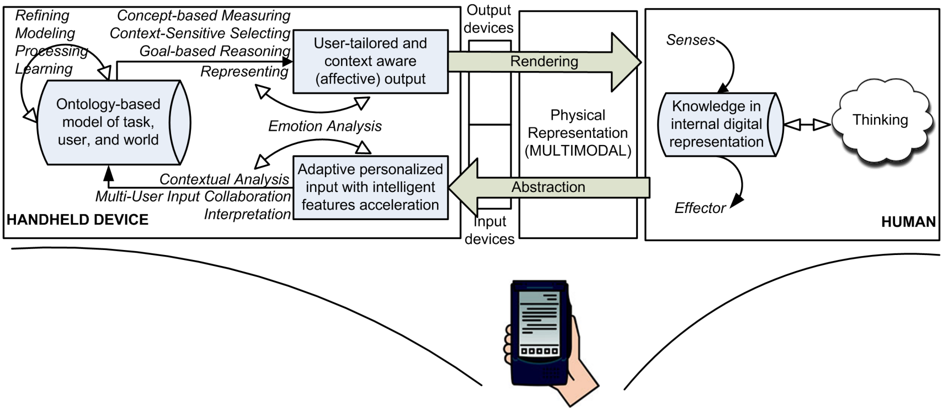

Refining
Modeling
Processing
Learning
Concept-based Measuring
Context-Sensitive Selecting
Goal-based Reasoning
Representing
User-tailored and context aware (affective) output
Ontology-based model of task, user, and world
Emotion Analysis
Contextual Analysis
Multi-User Input Collaboration
Interpretation
Adaptive personalized input with intelligent features acceleration
HANDHELD DEVICE
Output devices
Input devices
Rendering
Abstraction
Physical Representation (MULTIMODAL)
Senses
Knowledge in internal digital representation
Thinking
Effector
HUMAN

Figure 3.22: *A human handheld-device interaction on an adaptive user interface*

The variety of implemented input modules allows us to apply the framework in different domains (Figure 3.23). The language-based features of an input module can assist the module to adapt its behavior, which suits the user's activities and context of use learning from its experience with the particular user. Fast interaction with high accuracy input is mainly the goal of this module. Additionally, the learning component of the module updates the personalized library of the input prediction using data from the user's inputs (during interaction) and personal document storages (off line). As a consequence, the input interface becomes more intelligent and more flexible.

The output is typically divided into two stages: deciding what to communicate, and deciding how to communicate it. The first stage is called information generation, which includes the selection of communicative act to pursue, the selection of

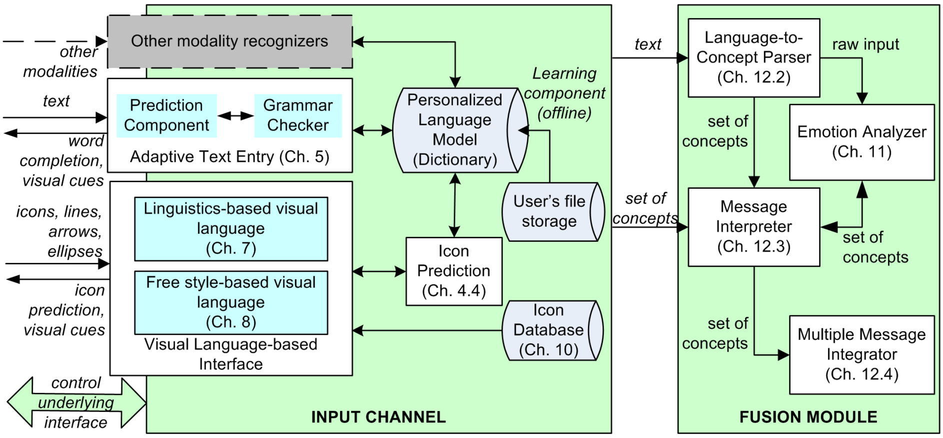

other modalities
Other modality recognizers
text
word completion, visual cues
Prediction Component
Grammar Checker
Adaptive Text Entry (Ch. 5)
icons, lines, arrows, ellipses
icon prediction, visual cues
Linguistics-based visual language (Ch. 7)
Free style-based visual language (Ch. 8)
Visual Language-based Interface
control underlying interface
Personalized Language Model (Dictionary)
Learning component (offline)
User's file storage
Icon Prediction (Ch. 4.4)
Icon Database (Ch. 10)
INPUT CHANNEL
text
set of concepts
Language-to-Concept Parser (Ch. 12.2)
raw input
Emotion Analyzer (Ch. 11)
set of concepts
Message Interpreter (Ch. 12.3)
set of concepts
set of concepts
Multiple Message Integrator (Ch. 12.4)
FUSION MODULE

Figure 3.23: *A schematic view of the architecture of the input processing*

propositional content, and the construction of discourse structure. The IM module works based on predefined interaction strategies. Each strategy provides some possible communication acts that can be selected depending on the user's current observed emotion. To select an appropriate communication act, the IM module evaluates the current contextual information of the user, the task and the world and selects a strategy based on predefined interaction goals. Together with a selected communication act from the IM module, the fission module adapts the current world model and presents information for a specific user. The second stage of generation is called information presentation. It consists of natural language generation and visualization generation. At this point, in our scenario, all users receive the most up-to-date information on their communication interface. The way information is presented on the interface, is highly tailored to each individual user's context, emotional state and available modalities. To support the rapid changes in the environment, the information generation and presentation can be triggered by the change in the internal knowledge content.

Adaptive Knowledge Content

Accurate mental models are one of the prerequisites for achieving context awareness (Endsley, 1995). A model can be described as a set of well defined, highly organized yet dynamic knowledge structures developed over time from experience. In our view, three variables can influence the development and maintenance of context awareness. They are individual, task, and environmental factors. These models provide the primary basis for subsequent decision making and performance in the operation of complex and dynamic systems to ensure similar user system interaction across different individuals and available modalities.

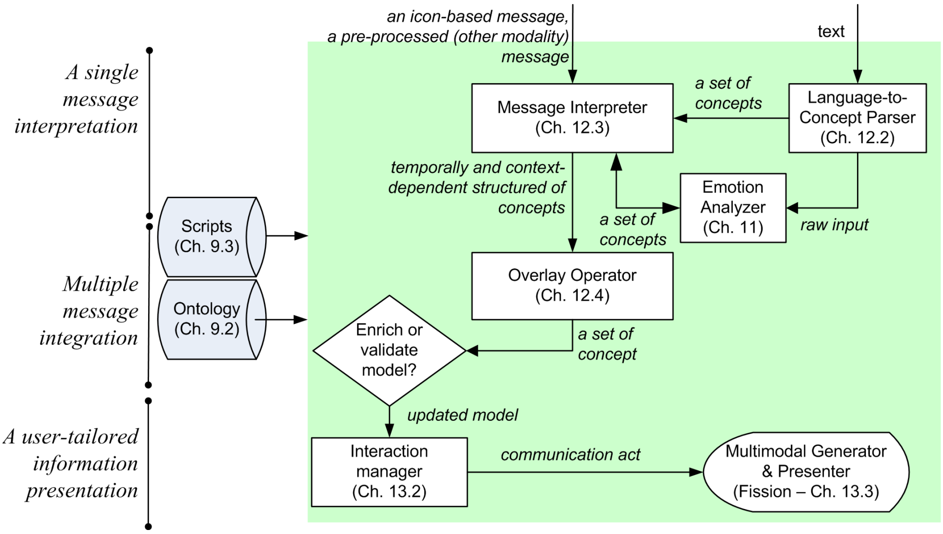

an icon-based message,
a pre-processed (other modality) message
text
A single message interpretation
Message Interpreter (Ch. 12.3)
a set of concepts
Language-to-Concept Parser (Ch. 12.2)
temporally and context-dependent structured of concepts
Emotion Analyzer (Ch. 11)
a set of concepts
raw input
Scripts (Ch. 9.3)
Overlay Operator (Ch. 12.4)
Multiple message integration
Ontology (Ch. 9.2)
Enrich or validate model?
a set of concept
updated model
A user-tailored information presentation
Interaction manager (Ch. 13.2)
communication act
Multimodal Generator & Presenter (Fission – Ch. 13.3)

Figure 3.24: *A schematic flowchart of message interpretation*

The chain information from the user initiation and interaction can lead to a conclusion of the current state of the user, the task, and the world. In the abstraction process (Figure 3.22), the string of input is interpreted and transformed into a set of concepts, a representation that is independent from modality. By employing the ontology and the scripts, these input concepts are analyzed to construct coherent semantic structures of knowledge (Figure 3.24). The result of this process is a contextual and temporal structure of concepts. As context awareness is a dynamic construct, changing the knowledge is dictated by the actions of users, task characteristics, and the surrounding environment. The dynamic situations can enrich or validate the constructed knowledge from previous interaction. The knowledge includes the results of an independent emotion analysis module that can be applied to analyze the emotion loading of the user's input and/or of the reaction toward the user's input. The context awareness is achieved by perception, comprehension and projection of situational information, which includes temporal and spatial concepts. The updated structure of knowledge can be handed over to the IM module and may assist the fission module in forming feedback to the user, which is specific to the particular user, task and events occurring in a specific context and interest.

Chapter 4

Input Prediction for Adaptive Interfaces

In which studies in comparing prediction methods and in comparing common and personal dictionary, are described. An adopted word prediction method for visual language-based interface is presented.

User requirements for usability in mobile context due to the small size of handheld devices, challenge traditional input design. In demanding situations, such as walking and talking, where the user's attention cannot be devoted fully on inputting, improvement in the input method performance is highly desired.

4.1 Input Method Challenges

One of the challenges of a new input method is the user requirement on ability to use it without the need for extensive practice (Bohan et al., 1999). Speech has been expected to be a compelling alternative to typing for text input. Despite the progress made in speech recognition technology, however, a study of Karat et al. (1999) showed that the effective speed of text entry by continuous speech recognition was still far lower than that of the keyboard (13.6 vs. 32.5 word per minute (wpm) for transcription and 7.8 vs. 19.0 corrected wpm for composition). Both recognition and synthesis are generally language specific and the best recognition accuracy is achieved with speaker specific training. We must notice that speech is tightly coupled with one interface technology. Speech does not work without microphones and speakers. Therefore, manual pen-based text entry remains one of the dominant forms of user interaction on handheld devices. The pen input accommodates single-handed interaction to offer users freeing a hand for holding the device.

Handwriting is arguably the most intuitive input method for handheld devices. However, according to MacKenzie & Chang (1999), its current recognition technology is still around 87%-93% accuracy; while LaLomia (1994) reported that users are willing to accept a recognition error rate of only 3%. Although the entry rates can be improved to 97% after 3 hours of practice (Santos et al., 1992), human's hand entry speed is limited to 15 wpm (Card et al., 1983). Thus, the entry rates of handwriting can never reach those of typing, 20-40 wpm (Mackenzie & Soukoreff, 2002).

Word prediction is frequently used in writing support devices. Its purpose is usually to improve the text input speed and the quality of spelling and syntax. It predicts which word tokens are likely to follow a given segment of text. As a result, a few keystrokes produce complete words or word sequences and the number of keystrokes necessary to generate a text will be reduced. The word prediction operates by generating suggestions for possible words or word completions before the user has finished entering a word or sentence. The user either chooses one or continues entering characters into the interface until the intended word appears. A selected suggestion is automatically inserted into the text. Predictions always require at least one character to be able to predict the intended word. Proper prediction takes the previous word(s), as well as any initial substring of the intended word into consideration when predicting the next word.

A language model is a basis to predict single word and multiword phrases. Three main language approaches can be distinguished: (1) syntactical approach, which deals with syntactical rules and grammatical attributes (Guenthner et al., 1993); (2) statistical approach, which uses word classifiers that are derived from a probabilistic language model (Shieber & Baker, 2003), and (3) mixed method approach, which translates an input into a syntactical attribute string and computes the most likelihood of the syntactical attribute actual word (Fazly & Hirst, 2003). Recent attempts have been done on increasing word prediction's efficiency, such as: (1) a heuristic method by analogy with existing similar words (Boissiére & Dours, 1996) or by calculating two-highly correlated words (Matiasek & Baroni, 2003), (2) incorporating external information (such as time, emotional state, location and the like) during building the corpus (Lesher & Rinkus, 2001), and (3) specialized devices and interface (Kühn & Garbe, 2001). Most of the works of the last approach focus on designing displayed keyboard on the screen. For example, arranging the most likely characters to be selected next so that the user can select them more quickly.

4.2 Which Input Prediction Method?

The input task on a telephone pad is an example where an input prediction can be applied to accelerate the input speed. Since the size of a mobile phone is very limited and usually there is no place to represent every character by an individual key, a limited keyboard is used where every key represents more than one character. As a consequence there is no one-to-one correspondence between a string of key-pressures and a string of characters. Every key represents three characters so a string of keys is an ambiguous message and can represent more than one word. But not every random string of key-pressures represents (part of) a word. A limited and insufficient stored vocabulary, spelling rules and grammar can limit the amount of possible character combinations and may affect the performance of the prediction. In many cases the first pressed keys represent a unique word. Therefore, it is not necessary to type the whole word or the whole string of keys.

An experiment has been performed to analyze three language models underlying the design of a limited keyboard: (1) hash tables, (2) n-gram, and (3) Markov model. The experiment aimed answer the question: which input prediction method does perform better on a limited keyboard (such on a telephone pad) and with small size dictionary? This section discusses the analysis on the properties of each approach to disambiguation of the key-press sequence given some knowledge of the structure of the encoded content, such as station names corpus.

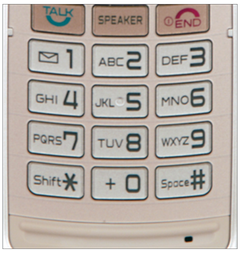


Figure 4.1: *The telephone pad showing informal mapping between the set of digits and the set of characters*

Methodology

In this experiment, we modeled the input task using a telephone pad on an OVR (Openbaar Vervoer Reisinformatie - Public Transportation Information Service) Internet site. Let $\chi = \{A, B, \ldots., C\}$ contain only 26 upper case characters and let $\mu = \{2, 3, \ldots, 9\}$ be the set of digits. A $\pi \colon \chi \mapsto \mu$ function gives the mapping from the character to the digit, as informally defined by the telephone pad model (Figure 4.1). Users who want to travel from a starting station to an end station have to type both names. Instead of typing twice or third times, in this simulation, the users will only need to press once for inputting the first, the second or the third character on a certain key. Since the received message is ambiguous, a prediction will compute the probabilities of the intended station name.

The Algorithm of the Prediction Methods

The prediction models used a corpus of 600 names of railway stations (including their aliases). The vocabulary is denoted as $\zeta \subseteq \cup_i \chi^i$. It is a subset of all names that can be formed from a string of characters of χ. ζ is built from a training corpus that includes the frequency of travelers start from or arrive at a station. If $k = \{k_1, k_2, \ldots, k_n\}$ is the vector of received keystrokes then this means $k_i \in \mu$ for $1 \leq i \leq n$. Let $c = \{c_1, c_2, \ldots, c_m\}$ be the vector of characters from χ that is a station name from the vocabulary ζ. Therefore, a key vocabulary $\xi(k)$ for a given $k \in \mu^n$ is defined as:

$$\xi(k) = \prod_{j=1}^{n} \pi^{-1}(k_j) = \prod_{j=1}^{n} \{n | \pi(n) = k_j\}$$

This ξ function returns a set of all the station name vectors of length n that can be formed using the characters that correspond to appropriate keys k_j. In addition, a sub-vector $x_{j:i}$ where $i < j$ from a vector $x = (x_1, \ldots, x_n)$ is defined as $x_{j:i} = (x_i, \ldots, x_j)$.

Each prediction method is described in the following.

- *Hash tables method.* The model depends on the existence of vocabulary ζ, which is taken from the representative text containing station names. An ordered tree data structure, a trie, is utilized, in which the out-degree of each node is less or equal to the number of elements in the digit set ($|\mu|$). The trie encodes all the station names in ζ (Figure 4.2). Each node of the trie can store a `WordList`. The coding is as follows. The links between the nodes of the graph are each weighted by one (and only one) element of the keystroke character μ. The nodes are stratified into levels. The path from the root node to a node on level l contains a sequence of links $(k_1, \ldots, k_l)$ and corresponds to a sequence of l key-presses. The invariant of the trie is: every node in the a subtrie on level l, corresponding to a list of key-presses k, has the station names in their word lists whose characters map by function π to the given vector k. If a particular part of the subtrie is empty, because no names correspond to a particular path, the coding is pruned.

  The searching function is performed as follows. For an input vector k, the search is started at the root and the only node inhabiting is $l = 0$. The function examines each element of the vector and follow the link labeled with the corresponding label to a node on $l = 1$. The algorithm is repeated with an input sequence of $k_{n-1:1}$ until it returns a matching name or null (if the link has no descendants).

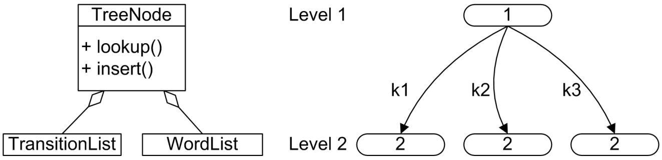


Figure 4.2: *The structure of the trie node and a view at a trie*

- *N-grams method.* In the n-grams approach, instead of storing all words, the model only stores the joint probability distribution of a vector of n characters. For a given n we have an n-gram: unigram for $n = 1$, bigram for $n = 2$, and trigram for $n = 3$, and so forth. The joint probability distribution for an n gram is given by $P(c)$, where $c \in \chi^n$. This function is obtained empirically by analyzing a representative sample station names. As $P(c)$ is formed, the model does not work anymore with ζ. This experiment was limited only for $n = 1, 2, 3$. If the posteriori probability that given k of length l and the input c (typed in), is maximized, then c^* is denoted as the best match according to this criterion. The max function requires iteration over the whole l-dimension space of the input sample to compute the inverse maximum. The search space can be reduced by noting that it only needs to examine such c that can be obtained from rolling back from k to c using the function $\xi(k)$. The conditional probability is proportional to the marginal $P(c|k) = 1/P(k) \cdot P(c,k)$. This holds as long as $c \in \xi(k)$ is true.

Therefore:

$$c^* = arg \max_{c \in \chi^l} P(c|k) = arg \max_{c \in \xi(k)} P(c|k) = arg \max_{c \in \xi(k)} P(c)$$

$$P(c) = P(c_1, \ldots, c_n) = \begin{cases} \prod_{i=1}^{n} P(c_i) & \text{for unigram model} \\ P(c_{2:1}) \prod_{i=2}^{n-1} P(c_{i+1}|c_i) & \text{for bigram model} \\ P(c_{3:1}) \prod_{i=2}^{n-2} P(c_{i+2}|c_{i+1}, c_i) & \text{for trigram model} \end{cases}$$

The conditional probabilities of the bi- and trigram models can be expressed in terms of joint probabilities, which are known a priori:

$$P(c_{i+1}|c_i) = \frac{P(c_{i+1}, c_i)}{P(c_i)} \text{ and } P(c_{i+2}|c_{i+1}, c_i) = \frac{P(c_{i+2}, c_{i+1}, c_i)}{P(c_{i+1}, c_i)}$$

All the unconditional probabilities can be gathered from the frequency data of the unigrams, bigrams, and trigrams.

As $P(c_i)$, $P(c_{i+1}, c_i)$, and $P(c_{i+2}, c_{i+1}, c_i)$ values are extracted from ζ, the probability of all station names can be maximized over attainable by inputting k and obtaining c^*. All terms of the products are independent and can be maximized individually to obtain the global maximum. The sequence obtained in this way is the most probable n-gram sequence.

* *Markov model with Viterbi.* In this experiment, we used the Markov model for obtaining the component backwards: starting from the expression that we wish to maximize using the conditional probability expression below:

$$c^* = arg \max_{c \in \xi(k)} P(c|k)$$

$$P(k|c) \equiv P(k_{n:1}|c_{n:1}) = P(c_n, c_{n-1:1}|k_n, k_{n-1:1})$$

where $P(k|c)$ emphasizes the n-th element of both the word an input vectors. It yields the most likely set of transitions to produce the keyword k. The idea underlying the following few lines is to express this conditional probability in terms of pre-computable values and a smaller instance of itself. If the last equation is reshuffled to eliminate the dependency upon k_n and the conditioning on k_n upon $k_{n-1:1}$, then:

$$\begin{aligned}
P(c_n, c_{n-1:1}|k_n, k_{n-1:1}) &= \frac{P(c_n, c_{n-1:1}, k_n|k_{n-1:1})}{P(k_n|k_{n-1:1})} \\
&= \frac{1}{P(k_n)} \cdot P(c_n, c_{n-1:1}, k_n|k_{n-1:1}) \\
&= \frac{P(c_n, k_n|c_{n-1:1})P(c_{n-1:1}|k_{n-1:1})}{P(k_n)} \\
&= \frac{P(k_n|c_n)P(c_n|c_{n-1:1})P(c_{n-1:1}|k_{n-1:1})}{P(k_n)} \\
P(k_n|c_n) &= \sum_{c\in\pi^{-1}(k_n)} P(c|c_n), P(c_n|c_{n-1:1}) \\
&= P(c_n|c_{n-1})
\end{aligned}$$

Therefore:

$$\begin{aligned}
P(k|c) &= P(k_{n:1}|c_{n:1}) \\
&= \frac{P(c_n|c_{n-1})}{P(k_n)} \cdot \sum_{c\in\pi^{-1}(k_n)} P(c|c_n) \cdot P(c_{n-1:1}|k_{n-1:1})
\end{aligned}$$

where $P(k_{n:1}|c_{n:1})$ is expressed in terms of $P(c_{n-1:1}|k_{n-1:1})$ and $P(c_n|c_{n-1})$, which are pre-computable quantities as transition probabilities between the states the Markov model. $c_i \in \chi$ is the states of the model with $k_i \in \mu$ as transitions. On the recursion, the priori probabilities of states of the expression is set to $\pi_c = P(c_{1:1}|c_{0:1}) = P(c_1)$ for each of the possible $c_1 \in \chi$. In this way, the transition probabilities can be given in terms of the bigram, which has been computed in the n-gram method. The Viterbi approach can be used to compute c^*, the global maximum using the algorithm that exploits the recursion.

Result Analysis

The hash tables method yields a structure that is built once. It is suited for fast searches of words that match a particular input sequence keystrokes as long as the vocabulary is sufficiently large. To alleviate this shortcoming, an alternative approach was analyzed, where the vocabulary is only used to build a fixed size table. The experimental results suggested that it might be possible to provide a good decoding that only exploits local dependence of the characters within a word, but that it depends heavily on the text used for training. The searching only need to traverse a single path down the trie. It is also easy to obtain all the words matching the given prefix by finding a node whose path from root encodes the prefix and then getting all the words from the word lists of the subtrie induced by that node. Adding an new word into the vocabulary is also convenient by mapping the word onto a corresponding key-press vector with function π. The new word is then inserted into the word list of the node that is the furthers down the search path.

The disadvantage of the hash tables approach is that at all times we need to have a reference vocabulary at hand. This may be a formidable constraint if we want to apply the method to other domain. As a large overhead and volume are present at each node and overall structure of the trie, the encoding is not efficient. Furthermore, the approach needs an equivalent storage space to host a trie on a memory device. Although the size increment will surely decrease as the trie is filled with more and more words, but the initial growth may make it unmanageable on a handheld device.

At design phase, no vocabulary is needed and the storage space is fixed once the n of the n-gram model is determined. The method creates the n-gram dictionary from any input (training set) text at run time. For a given n, using the n-gram method, at most $|\chi|^n$ values are stored. This is considerably less that what is consumed by a trie on the hash tables method. It was assumed during the experiment that the decoding quality would increase with enlarging the training set, however, we could not conclude whether a saturation point exists that yields significant decoding performance boost.

A Markov model is an elegant way to encode the character interdependencies. This model allows us to express the probability over the entire sequence. However, as already showed, to build this model need unigram and bigram data. Similar to the n-gram method, the model can be constructed even with a small size dictionary.

4.3 Personalized Dictionary, Is It Necessary?

In contrast to physical keyboards, with on-screen touch keyboards, the key layout has a major effect on the text entry performance (Isokoski, 2004). This is because typing is strictly sequential. To type a character, user has to move the pen from one key to the next and during this time there can be no preparation for the following key. Thus, an input prediction that is able to collect knowledge about user linguistics compositions and use the knowledge to alter its future interaction can greatly enhance text entry performance. An experiment on a comparison of common and personalized dictionaries proved this. The experiment was aimed to answer: (1) can text entry performance be improved by using a personalized dictionary? and (2) which and how personal data should be used adaptively? It used the following datasets:

1. Common English corpus from British National Corpus (BNC), which consists of 166261 words (BNC, 2007).

2. 4.4 MB personal documents, such as documents, spreadsheets, and schedules, which consist of 19121 words. The author is a researcher in the field of multimodal communication.

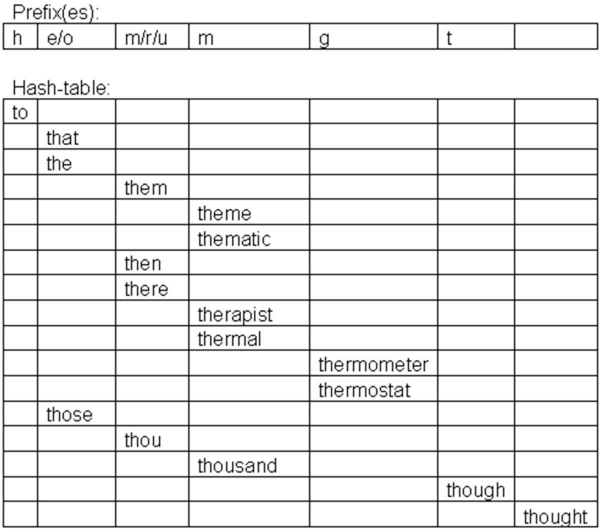
3. 7.2 MB corporate e-mails, which consist of 13046 words (Corrada-Emmanuel, 2007), were taken from internal e-mails of the Enron corporation, an energy company in Houston, Texas.

4. 4.2 MB chat-logs, which consist of 15432 words (ZetaTalk, 2003), contain philosophical discussions about topics, such as afterlife or aliens presence.

Methodology

This experiment consists of two following parts:

1. The coverage of the BNC was compared to unigrams and bigrams in personal datasets. All words were collected from each dataset and their frequency were calculated. 5500 most frequent words were selected from each personal dataset. These words appeared at least 20 times in each dataset.



Prefix(es):

h	e/o	m/r/u	m	g	t	

Hash-table:

to						
	that					
	the					
		them				
			theme			
			thematic			
		then				
		there				
			therapist			
			thermal			
				thermometer		
				thermostat		
	those					
		thou				
			thousand			
					though	
						thought

Figure 4.3: *A part of hash tables for the first character "t" (schematic view)*

2. The word completion was simulated without any statistical model using hash tables. Using this method, how many appropriate number of character entries are necessary before a user can select a completion could be analyzed. Figure 4.3 shows the character entries to serve as a prefix before a completion. Different columns show that some characters are necessary for completing the word. For example, for "thermometer" needs "t", "h", "e", "r", "m", and "o" to distinguish it from "thermal".

Result Analysis

Table 4.1 shows that on average 87% of words in personal datasets and about 74% of the union of all personal datasets were covered by the BNC. Most words that were not covered by the BNC from personal documents were abbreviations, names and specific terms, such as "xml", "website", "lexicalized" and "wordnet" in the field of computer science. 78% of the words in e-mail datasets that were not covered by the

BNC were addresses and names of persons, products and organizations. Other 11% were specific terms, such as "teleconference", "worldnet" and "unsubscribe" in the field of communication network. Some of the words in chat logs that were not covered by the BNC were popular terms in chatting or informal conversation, such as "lol" (laugh out loud), "okidok" or "yup" (OK), "thingie" (such thing), "heck" (hell) and emoticons, for example: ":)" for smile and ":))" for laughing. Others (91%) were names and internet addresses.

Table 4.1: *The coverage of BNC toward the personal datasets*

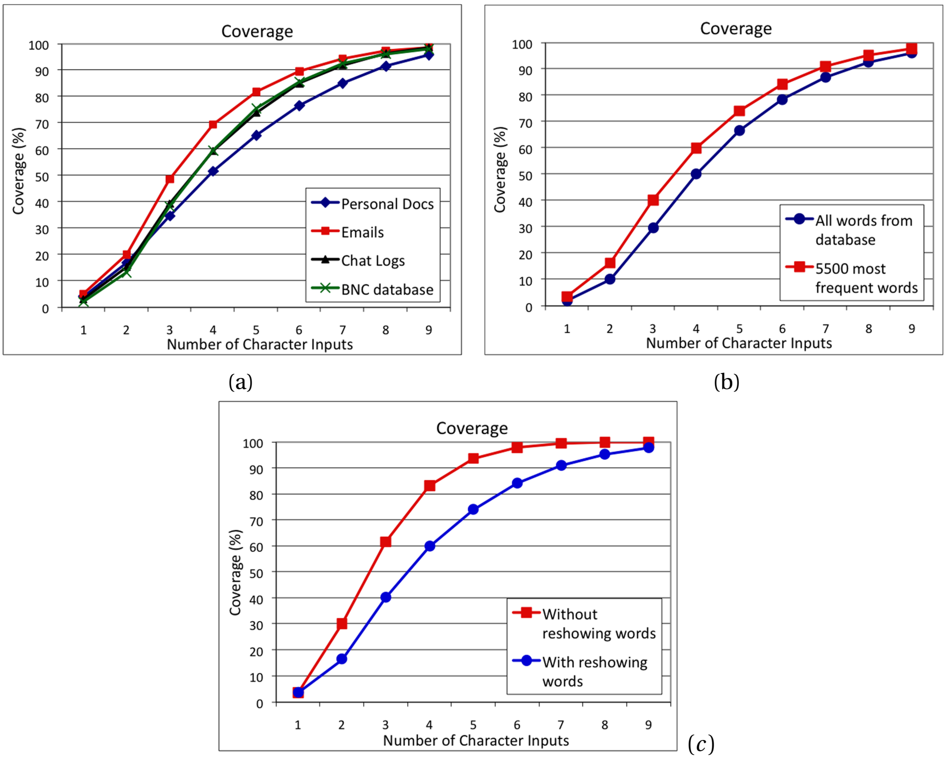
Unigram	**Number of Words**	**BNC Coverage (166261 words)**	$A \cup B \cup C$ **Unigrams Coverage**
A: Personal Docs	5500	4982(90%)	49%
B: E-mails	5500	4740 (86%)	49%
C: Chat Logs	5500	4754 (86%)	49%
$A \cap B \cap C$	1685	1674 (99%)	15%
$A \cup B \cup C$	11168	9579 (85%)	

Table 4.2 shows that the BNC had the lowest coverage for the personal document dataset. Although all words in each bigram were covered by the database, the compositions of them might not. Most of these bigrams were terminologies in a specific domain. For example: "human interaction", "usability testing", and "interface design" in the field human-computer interaction; "multimodal fission", "dialogue management" and "natural language" in the field multimodal system; and "emotion expressions", "facial recognition", and "muscle coordination" in the field nonverbal communication. They were considered as the most frequent bigrams (at least 29 times).

Table 4.2: *The coverage of BNC toward the personal datasets*

Bigram	**Number of Bigrams**	**BNC Coverage (726000 bigrams)**	$A \cup B \cup C$ **Bigrams Coverage**
A: Personal Docs	54829	33994 (62%)	56%
B: E-mails	10505	7016 (83%)	11%
C: Chat Logs	36801	29809 (81%)	37%
$A \cap B \cap C$	2426	2348 (96%)	2.4%
$A \cup B \cup C$	89275	68742 (77%)	

Most bigrams in the e-mail dataset that were not covered by the BNC were terminologies in the corporate domain, such as "financially bankrupt", "employee transition", "expense report" and "retirement plans". Small amount bigrams were in the field of communication, such as "intended recipient", "conference call", and "video connection". The chat logs also contained bigrams in a specific domain that were not covered by the BNC, such as "planet x", "pole shift", and "star children". Small amount bigrams were about science, such as "gravity particle", "volcanic ash" and "orbital path". The experimental results showed that user personal word usage had a strong correlation with the user's task context. The coverage of the BNC to the intersection of the personal datasets was quite high. However, the personal datasets shared only a small amount. The reason could be that the datasets were from a specific context and/or not from the same source.



(a)

(b)

(c)

Figure 4.4: *(a) The coverage of 5500 most frequent words from four datasets, (b) the average coverage of all words versus 5500 most frequent words from four datasets, and (c) the average coverage of the 5500 most frequent words from four datasets with re-showing words and without re-showing words*

Figure 4.4(a) shows that on average 3.6% of the cases, a user was able to select an intended word in just one entry. Almost similar coverage in all datasets occured for every prefix. If we assumed that the completion is using all words in datasets, the results showed that the performance of the completion was degraded due to the inclusion of lower frequency words (Figure 4.4(b)). Figure 4.4(c) shows if the completion was not re-showing the same completion once these words were shown for a given word being entered. For example, when "ther" is written, "there" is one possible completion. If "e" is inputted next, a better option is to show a different word completion, for example "thereby". In this way, those empty cells, for example from "there" to "thereby" and from "thermo" to "thermometer", are disappeared.

This study found that the word completion showed better performance using a relatively small size dictionary containing the most frequent words, which was also shown in the previous finding that the personal datasets shared only a small number of the corpus. The experimental results indicated that a personalized context-based dictionary could offer a more efficient word completion than a large common dictionary. They could also imply to the accuracy of the word prediction if syntactically implausible words were also excluded from its prediction space. Therefore, besides saving time and energy in inputting, the prediction can also assist the users in the composition of well-formed text. In addition, the number of user inputs for a desired word can be reduced if the system takes an assumption that a suggested word is rejected after the user selects the next character. Such an option will reduce the number of inputs to select a desired word, since users sometimes miss the initial appearance of the word they intended and enter more characters than necessary. Therefore, the user can have better language coverage since each suggestion word is shown only once. This last finding is coherence with Wobbrock and Myers (2006).

4.4 Input Prediction for Visual Language

In a visual language-based interface, visual symbols, such as icons, may replace a word, a phrase or even a sentence. Due to the high number of presented icons on the interface, scrolling on the interface is inevitable. The icons usually are displayed on and can be selected from a menu. Extra steps to find intended icons may also be needed. An icon prediction could help the user to eliminate these steps. Besides for improving the input speed, in creating visual language-based messages particularly, the prediction also improves the quality of syntax. Using a language model, the prediction can predict which icons are most likely to follow a given segment of a visual language-based sentence that is grammatically correct. As a result, the number of icon look-up processes that necessary to generate the string will be reduced.

To build input prediction for a visual language-based interface, a word prediction technique can be adopted. The prediction operates by generating a list of suggestions for possible icons after the first icon is selected. A user either chooses one of the suggestions or continues entering until the intended arrangements of icons appear. A selected suggestion is automatically inserted into the input area.

The probability of the icon string in a visual language-based sentence is estimated as the product of conditional probabilities. To predict the most likely icon in a given context, a global estimation of the icon string probability is derived which is computed by estimating the probability of each icon given its local context (history). Here, estimating conditional probabilities of n-grams type features are used. The conditional probabilities of the bigram and trigram models can be expressed in terms of joint probabilities, which are known a priori:

$$P(w_{i+1}|w_i) = \frac{P(w_{i+1}, w_i)}{P(w_i)} \text{ and } P(w_{i+2}|w_{i+1}, w_i) = \frac{P(w_{i+2}, w_{i+1}, w_i)}{P(w_{i+1}, w_i)}$$

where w_i is an icon in an inputted visual language-based sentence.

To compute multi-grams model, the developed visual language-based interface needs to collect the data from user selections during the interaction. Unlike a sentence in a spoken language, which consists of a sequence order of words, a sentence in a visual language is constructed by an arrangement of icons. Besides a spatial property, the icon may have a temporal property and relations to other icons. The relation may be undirected or directed. In the current version of the visual language-based interface, we only concern the temporal property and these relations.

A bigram corpus can be denoted as $bigram(w_i, w_{i+1}) = bigram(icon_p, icon_q)$ where $p, q \in [1 \ldots n]$. An *undirected relation* of two icons is distinguished by their temporal property. This means w_i is inputted earlier than w_{i+1}. These icons may be connected (using a non-arrow connection) or grouped. A *directed relation* of two icons in the bigram corpus is distinguished using two terminologies: "topic" and "comment". The comment is the icon that explains the topic. If both icons are connected using an arrow then the comment is the begin of the arrow and the topic is the end of the arrow ($comment \longrightarrow topic$). Based on this relation, we have:

$$bigram(icon_{topic}, icon_{comment}).$$

The trigram corpus is constructed using two related bigrams. Therefore, a trigram corpus is computed as:

$$\begin{gathered} bigram(icon_p, icon_q) \\ bigram(icon_q, icon_r) \\ trigram(w_i, w_{i+1}, w_{i+2}) = trigram(icon_p, icon_q, icon_r) \end{gathered}$$

where $p, q, r \in [1 \ldots n]$.

Chapter 5

Text Entry with Language-based Acceleration

In which an adaptive on-screen keyboard for a pen-based interaction is described. It combines tapping-based and motion-based text input with language-based acceleration techniques, including personalized and adaptive task-based dictionary, frequent character prompting, word completion and grammar checker with suffix completion.

Inspired by the results of the experiment described in the previous chapter, an on-screen keyboard that offers an easier and faster method of entering text with a pen on handheld devices, is developed. A method for adapting its predictive ability according to personal word usage of the user, context and syntax rules has been developed too. Frequently used characters are presented to the user in different key sizes and color contrasts according to their relative probabilities to facilitate visual searching.

5.1 Text Entry Technologies

In practice the most popular pen-based keyboard design is still the QWERTY layout and its language-specific adaptations. It has been observed that this layout is not optimal for pen-based text entry because the distance between common adjacent characters is too far (Mackenzie & Soukoreff, 2002). Previous work in developing adapted keyboard layouts for handhelds and single-handed use has concentrated on alternative key configuration for improving entry speed. Some alternative keyboard layouts (other than Qwerty) with movement minimizing have been developed. The Cirrin (Circular Input - Figure 5.1(c)) arranges the characters inside the perimeter of an annulus (Mankoff & Abowd, 1998). The most commonly used digrams (two successive letters) are nearest to each other. Therefore, distances traveled between characters are shorter than Qwerty. However, since there is not any predictive feature, a user must attend to the interface when entering text.

Typically, there are three types of text entry methods. First, tapping-based text entry, in which the pen must be tapped (clicked) for selecting characters. It requires intense visual attention, virtually at every tap, which prevents the user from focusing attention on text output (Zhai & Kristensson, 2003). An example of these keyboards is Fitaly (Langendorf, 1988) (Figure 5.1(a)). Second, motion-based text entry interprets informal pen motions as character inputs, for example Cirrin. Finally, a hybrid type augments tapping and motion, for example IBM Atomik Shorthand (Zhai & Kristensson, 2003) (Figure 5.1(b)).

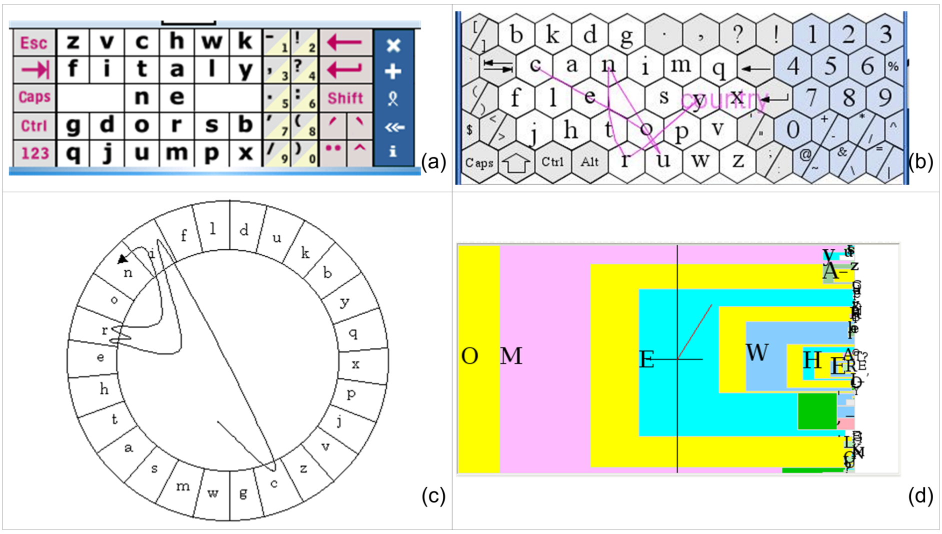


Figure 5.1: *Examples alternative on-screen keyboards: (a) Fitaly, (b) IBM Atomik Shorthand, (c) Cirrin and (d) Dasher*

Zhai and Kristensson reported that visually guided tapping is easier for novice users (Zhai & Kristensson, 2003). Since simple tapping movement may feel tedious to repeat for prolonged use, motion-based input is preferred by experts. Some text input techniques have been developed with both minimizing movements and predictive features. For example, Dasher (Figure 5.1(d)) uses prediction by partial matching, in which a set of previous symbols in the uncompressed symbol stream is used to predict the next symbol in the stream (Ward et al., 2000). Most word predictions have been developed based on n-gram method, which often suggest syntactically implausible or excluding more plausible but lower probability from its suggestion list. This can confuse users by inappropriate suggestions. A small amount of improvement on word prediction can be achieved by using syntactic information in the prediction, such as POS n-gram information, since statistical models are considered weak in capturing long-distance co-occurrence relations between words (Garay-Vitoria & Abascal, 2004). Another way is by excluding implausible or ungrammatical words from the prediction's input (Wood & Lewis, 1996). Most of these grammar checkers employ a POS tagger and a set of pattern matching rules (Heidorn, 2000). Although a prediction can improve entry performance, Anson et al. (2006) reported that searching through its word list is considered as tedious and disruptive.

Dasher is an adaptive on-screen keyboard for both single and zero-handed users. It employs continuous input by dynamically arranging characters in multiple columns positioning the next most likely character near the user's pen input. The options are presented to the user in boxes sized according to their relative probabilities, to optimize the movement time. Dasher demands user's visual attention to dynamically react to the changing layout. Although the developers claimed that Dasher needs short training time, its user text entry rate is less than QWERTY layout's. Typical writing errors were spelling and syntax errors.

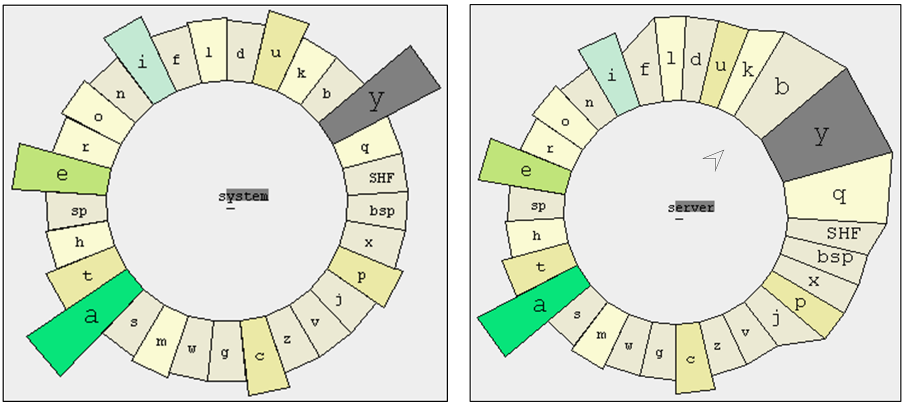
An interview with a Dasher user, who is a computer science student and suffers from cerebral palsy (impairs physical movement and limits speech) has been performed (Fitrianie & Rothkrantz, 2007a). To enable to communicate he uses a computer device and Dasher for two years with a head-tracker device. The only reason is because Dasher is a motion-based text entry. It was reported that the boxes sizes and color contrasts on its word prediction are very important as visual cues for next character selections. However, the character arrangement constantly changes makes the user dizzy after some time. Moreover, it is not always easy to correct errors, since the interface does not provide a fast error recovery button/menu. The current implementation helps the user in writing text and documents, but is less suitable for writing in specific context, such as daily talks, e-mailing, emergency noting and programming. It is desirable to have such a text entry device that works in specific domains with a personalized vocabulary.

5.2 The Adaptive Cirrin

Figure 5.2 shows the adaptive Cirrin. The interface gives visual cues, such as different key sizes and color contrasts, for the next-character and the next-word selections, without changing the character layout. A standard on-screen keyboard does not fit with this specification. On a keyboard whose characters are arranged on the circumference of a polygon or a circle or in two parallel columns, it is possible to expand the keys' size. Therefore, the Cirrin layout (Mankoff & Abowd, 1998)(Figure 5.1(c)) was adopted, which is based on a scoring function to calculate the most used adjacent characters. It contains 26 English characters. The new design differs from the original Cirrin in the following five aspects.

Geometry. The middle of the ring is an input area, where selected characters of a single word are displayed. To support direct perception of the user, each selected character will be displayed in the text area directly, where a message is composed. The visual cue on a key gives information about the likelihood of the next character selection. The current implementation uses 200% expansion and the highest contrast color for the highest probability characters. The key of lower probability characters is expanded and colored based on its proportion to the highest probability characters.

Character Set. Two characters, space and backspace, are added to the ring to support a quick error recovery. In the event of an erroneous correction, the user can make a backspace stroke or press the backspace key to undo the selection and restore the completion as it appeared before. This makes completions quickly undo-able. An additional matrix can be placed on the right side for numbers, return, control, period,


system
server
i f l d u k b y q SHF bsp x p j v z c g w m s a t h sp e r o n

(a)

(b)

Figure 5.2: *(a) The adaptive Cirrin and (b) the adaptive Cirrin with fisheye style*

punctuations, comma and other additional keys.

Input Style. The adaptive Cirrin allows both tapping and motion-based input and combination of them. When entering a word, a user may begin with any mode and continue with another, or only one of them. New selections will be appended to the previous selections. In the motion mode, dragging starts and ends in the middle of the circle. When the user stops dragging, a space will be added at the end of the word. When a space is selected, the input will be flushed to the text area. Deleting a space will return the previous inputted word back to the middle of the circle.

Word Completion. As the user enters each keystroke, the adaptive Cirrin displays most likely completions of a partially typed word in the input area. It indicates which characters of the word are not yet selected. As the user continues to enter characters, the suggestion will be updated accordingly. The special feature is that the adaptive Cirrin only shows a suggestion once. If a suggestion is turned down, the completion will show another suggestion, which may have a lower probability. A suggested word is assumed to be rejected if the user selects a new character instead of selects the suggestion. The user can select a word completion with a single tap (in tapping mode) or a left-to-right line motion (in motion mode) in the middle of the circle. This word will be flushed to the text area and a space is added next to the new word.

FishEye Style. As the cursor moves from and to a certain key, in the fisheye mode (Figure 5.2)(b), the nearest keys are expanded based on their probability and distance to the cursor, which is predicted using the function `forward_viterbi` (Figure 5.3), which takes the following arguments: (1) the sequence of observations (the most probable words); (2) the set of hidden states (the most probable letters based on the distance to the user's cursor); (3) the start probability of each letters; (4) the transition probabilities of each letters after a new letter is selected; and (5) the emission probabilities of each words after a new letter is selected. The algorithm works on the mapping of the probability of the selected path of the current state to the next state. The selected path is computed as the corresponding letter with the most probable value.

```
T = {}
for state in states:
    T[state] = (start_p[state], [state], start_p[state])
for output in observation:
    U = {}
    for next_state in states:
        total = 0
        argmax = None
        valmax = 0
        for source_state in states:
            (prob, v_path, v_prob) = T[source_state]
            p = emit_p[source_state][output] * trans_p[source_state][next_state]
            prob *= p
            v_prob *= p
            total += prob
            if v_prob > valmax:
               argmax = v_path + [next_state]
               valmax = v_prob
           U[next_state] = (total, argmax, valmax)
       T = U
# apply sum/max to the final states:
total = 0
argmax = None
valmax = 0
for state in states:
    (prob, v_path, v_prob) = T[state]
    total += prob
    if v_prob > valmax:
       argmax = v_path
       valmax = v_prob
return (total, argmax, valmax)
# end of function

# an example of input after the letter 's' is selected:
states = ('t', 'a', ..) #possible letters based on their distance to the cursor
observations = ('stamp', 'static', 'saturn', ...) #possible words
start_probability = {'t': 0.6, 'a': 0.4} #the probability of letters
transition_probability = {'t' : {'a': 0.7}, 'a' : {'t': 0.6}, ...}
emission_probability = {
   't' : {'stamp': 0.1, 'static': 0.7, 'saturn': 0.2},
   'a' : {'stamp': 0.6, 'static': 0.1, 'saturn': 0.3}, ...
}
```

Figure 5.3: *The Viterbi algorithm for predicting the cursor movement*

5.3 System Design

Figure 5.4 shows the class diagram of the adaptive Cirrin. The `FText` class handles the user system interaction. It has the `cirrin` module that contains the classes of the keyboard interface, such as the Key class and the InputArea class.

The Prediction class uses a dictionary that consists of uni-, bi- and trigrams and is stored in the ngrams module. The dictionary includes information about the POS tags and frequencies of each element. When the adaptive Cirrin is used at the first time, the Learning class parses all personal documents in the user's storage. The user may specify folders and files that can be extracted by this class. Otherwise, by default, it will extract first all personal documents and e-mails. This process fills the dictionary. The Learning class changes and adapts the dictionary using the inputs during interaction.

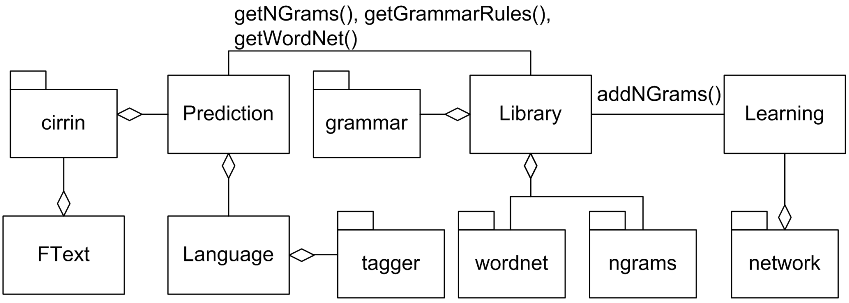


Figure 5.4: *The class diagram of the adaptive Cirrin*

The Prediction class operates by generating a list of suggestions for possible words after the first character is inputted. For the character input of the first word in a sentence, this class returns all words that start with the same set of characters. After the first word is inputted, the next possible words are predicted using a statistical approach that was derived from n-grams language model. The probability of a sentence is estimated using the product of conditional probabilities:

$$P(w_1, w_2, ..., w_n) = \prod_1^n P(w_i \mid w_1, ..., w_{i-1}) = \prod_1^n P(w_i \mid h_i)$$

where h_i is the relevant history when predicting a word w_i. To predict the most likely word, a global estimation of the probability is derived. The probabilities obtained from uni-, bi- and trigrams are weighted together using standard linear interpolation formula. The results of the prediction are ranked based on their probability. The POS information of a given word in the suggestion list is also included, since a word form may be ambiguous and adhere to more that one POS. This list is filtered to have all words that start with the same set of characters as the user's input.

Besides for improving the input speed, the adaptive Cirrin aims to improve the quality of syntax. Therefore, the overall motivation for the Language class is to enhance the accuracy of the prediction suggestions. This class does not by itself generate any prediction suggestions but filter the suggestions produced by the n-gram model.

The grammatically correct word forms will be presented to the user prior to any ungrammatical ones. The `Language` class excludes syntactically implausible words from the suggestion lists and includes suffix completions, in the following five steps:

1. Parse each sentence in the input using the `tagger` module and calculate the highest probability POS of each word. The POS tagged input is splited into chunks of phrase for detecting the cardinality and the tense of the sentence.
2. Create all (unavailable) forms for each word in the suggestion list. Currently, thirteen suffixes are used: "s", "ed", "er", "est", "ly", "able", "full", "less", "ing", "ion", "ive", "ment", and "nest". Using WordNet (Fellbaum, 1998), each new form is verified. All forms are added to the suggestion lists with the same probability.
3. Check each suggestion using the rules in the `grammar` module whether it is grammatical, ungrammatical or out of scope of the grammar. The ungrammatical ones are discarded from the lists. In the current implementation, classic BNF is used to define English grammar.
4. Choose the highest probable word from the dictionary.

5.4 User Testing

An experiment has been conducted consisting a small-scale user test in a laboratory setting. The aims of this experiment were to perform a test of the design concept and to address usability issues of the adaptive Cirrin. With the implemented adaptive Cirrin, we set up the experiment to research two questions: (1) what is the effect of using the on-screen adaptive Cirrin layout comparing to the on-screen Qwerty layout using pen? and (2) can different language-based features accelerate text entry and improve users performance? In this experiment, we tried to reduce the effect of learning Cirrin as a new text entry system in user performance by asking all participants to use the adaptive Cirrin for a certain period of time before taking the actual test.

5.4.1 Methodology

The experiment was performed in a small room, one table and two chairs. The test materials were: (1) a tablet PC with a pen and (2) an experimenter booklet and a participant task booklet. Only one experimenter assisted all participants during the experiment. The user interactions were logged and noted. The logger program unobtrusively gathers click stream data as users complete specified tasks.

Test Execution. The initial approach to the experiment was to read provided text. The text was simple. The participants were expected to memorize it after the first input. The text consisted of six sentences (each consists of 4-6 words; in total contains

29 words and 190 characters). They were taken from the developer's personal documents. This way allowed us to test the adaptive Cirrin in a reasonable simulation of typical usage and personalization where the language model had been trained on these documents.

Table 5.1: *The configuration of the adaptive Cirrin in the user testing*

Parameter	Configuration
Window size	440*640 pixels (portrait)
Number of keys	28 keys ([a...b], space and backspace)
Circle diameter	150 pixels for outer circle and 100 pixels for inner circle
Key's size	50 pixels normal length and 100 pixels maximum length
Key's angle	12.857 for normal size and 25.70 for maximum size
Input prediction size	10678 unigrams, 54829 bigrams and 85293 trigrams in the field of informatics.

For this experiment, the adaptive Cirrin was set with the configuration in Table 5.1. The online learning component was not available. Therefore, the language model had the same training samples for all sessions and all participants. During the experiment, the participants were expected to experience two aspects of usability issues on an on-screen text entry system: (1) usable layout design and (2) affordance of control (text entering and mistake correction). There were five experimental sessions:

1. The simple Cirrin (Figure 5.1(c)) that allows both tapping- and motion-based input.
2. The adaptive Cirrin (Figure 5.2(a)) with language-based features, such as personalized and adaptive task-based dictionary, frequent character prompting, word completion and grammar checker with suffix completion.
3. The adaptive Cirrin with language-based features and pruning suggestion (each suggestion is shown only once).
4. The adaptive Cirrin (Figure 5.2(b)) with language-based features, pruning suggestion and fisheye style.
5. The on-screen Qwerty layout.

To counter-balance learning affect, 4x4 Latin Square matrix was utilized for the first four sessions.

Questionnaire. At the end of each session, the participants were asked to fill in a questionnaire, which focuses on four aspects: (a) whether it is fast to select the intended character; (b) whether it is easy to make correction; (c) whether the interface design is usable; and (d) whether the language features can accelerate the input. The questionnaire contained 15-16 statements. The participants were asked to give rate

1-5 for each statement, such as 1: strongly disagree, 2: disagree, 3: neutral, 4: agree, and 5: strongly agree. At the end of the experiment, the participants were asked to order their preference of four interfaces.

Participants. Sixteen people took part in the experiment and one person as a pilot, recruited from the students and staffs of the Man-Machine-Interaction group, Delft University of Technology. We assumed that all participants had the same level of computer knowledge and experiences using a pen as an input device. In this experiment, there was no assumption on distinction of age, gender or cultural differences.

5.4.2 Measurement

The experimental results were quantitative and qualitative measurements.

Quantitative Measurement. Using the results of the logger, the following measurements were taken:

1. Text-entry speeds in terms of the characters per minute (CPM) and the words per minute (WPM).

2. Number of strokes per word. This referred to the number of key presses before inputting a new word including unnecessary extra characters and back space entries for the input correction.

3. Number of completion selection per word.

4. Text-entry errors, such as unnecessary extra characters, selecting neighbor error, incorrect completion.

Qualitative Measurement. To know more about the usability aspects of the adaptive Cirrin, the questionnaire and remark from each participant were analyzed. The focus was on the adaptive text entry features, such as: (1) the interface layout, (2) the frequent character prompting, (3) the word completion, and (4) other language-based features, such as the pruning suggestion and the fisheye style.

5.4.3 Result Analysis

The results of the study are discussed in the following three parts: input performance, usability assessment, and user satisfaction.

Input performance

Most of the test users only used regular keyboard for text entry. Only four participants were the user of on-screen keyboard with a pen input for relatively a small amount of time per day. We consider this was the reason of why the input speed of our Qwerty users was less than of the results from Mackenzie & Soukoreff (2002) (20-40 wpm). However, the overall experimental results showed a good comparison of the performance of our test participants on all tested keyboard designs. The results could be used to answer the research questions of this experiment.

Table 5.2: *Text entry speed*

Text Entry System	**CPM**	**WPM**	**#Strokes per Word**
Qwerty	103.92 ± 22.78	15.38 ± 2.87	8.01 ± 0.80
Simple Cirrin	39.57 ± 11.58	5.59 ± 0.85	7.55 ± 0.51
Adaptive Cirrin	70,00 ± 19.26	10.55 ± 2.99	5.11 ± 0.48
Adaptive Cirrin with Pruning Suggestion	79,11 ± 18.87	11.92 ± 2.54	4.59 ± 0.93
Adaptive Cirrin with Pruning Suggestion and Fisheye Style	53.28 ± 14.29	7.92 ± 2.00	5.64 ± 2.10

Table 5.2 shows the average text entry speed performed by the test users. The experimental results showed that text entry using Qwerty was still the fastest despite of the highest number of input strokes. The number of strokes per word includes the strokes for the correction task. The adaptive Cirrin with pruning suggestion was the fastest among other types of Cirrins. Moreover, the participants showed better performance in all adaptive Cirrins than in the simple Cirrin.

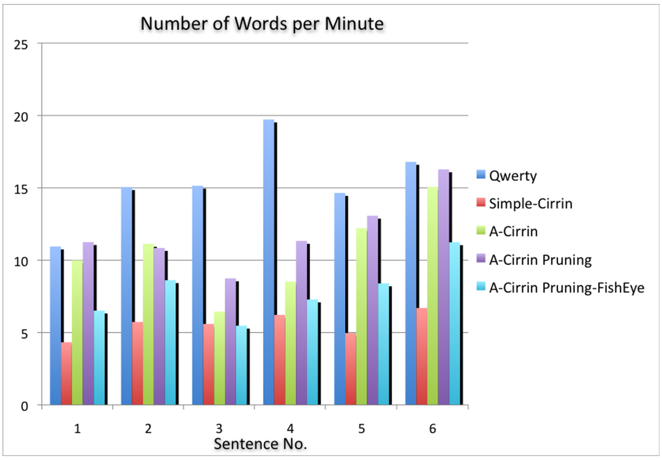


Figure 5.5: *The number of words per minute (for sentence 1 - 6)*

The experimental results also indicated that the best performance of the adaptive Cirrin with pruning suggestion users is still in the range of the input speed for Qwerty.

As shown in Figure 5.5, for sentence 1, 5 and 6, these users had almost similar performance as the Qwerty users.

Usability Assessment

Almost all remarks of the participants pointed into the same direction. The most important points of the usability assessment are summarized in three following aspects.

- *Keyboard design.* Most participant mentioned that the Cirrin layout was easy to learn. They found it easy to select their intended character. One user even was quite surprised with the accuracy level of her input on using a keyboard she never used before. Only 2% of the participants asked for redesigning the space position. These users found it disadvantage if they wanted to input "t", "h" and "e". Two other participants asked for a bigger key size.

  All participants indicated that it was easy and fast to correct input mistake and that the visual cues with different color and key sizes did not make them dizzy. The changing of displays did not make them loose their orientation either. Although the results showed that the different color did not help the users to find the character they intended, however the highlight made them aware of those keys. The participants tended to search the character from the highlighted keys first before other keys. Moreover, expanding the area of a key made it easier to select the key.

Table 5.3: *Completion task*

Text Entry System	**#Chars before Completion**	**Selecting Incorrect Completions per #Words**	**Input Words using Completion per #Words**
Adaptive Cirrin	4.30 ± 0.09	0.18 ± 0.01	0.80 ± 0.03
Adaptive Cirrin with Pruning Suggestion	1.35 ± 0.01	0.15 ± 0.03	0.81 ± 0.04
Adaptive Cirrin with Pruning Suggestion and Fisheye Style	1.55 ± 0.15	0.20 ± 0.23	0.87 ± 0.04

- *Language-based features.* All participant found that the prediction was very useful. Table 5.3 shows that 82% of words in the test case were inputted using the word completion. The words were inputted mostly after 1-4 key strokes. With this feature, the participants could input the text very fast. The results also showed that the participants selected less characters before the completion in the adaptive Cirrin with pruning suggestion than using other types of Cirrins. As mentioned before, the experimental results also showed that the frequent character prompts help the users to find their intended character in a fast and easy way. In addition, it was advised not to show the prediction results for the first word of a sentence since it may be pointless.

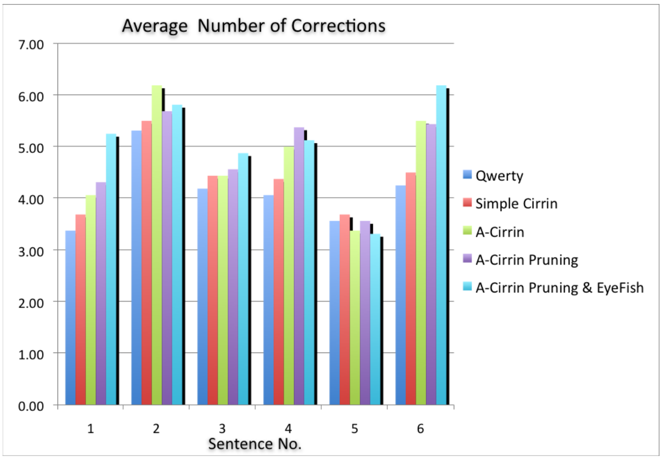


Figure 5.6: *The number of corrections (in sentence 1 - 6)*

- *Affordance of the controls.* It appeared that all participants had similar problems using the adaptive Cirrin with fisheye style. Although the frequent character prompts, the word completion and the pruning suggestion were still very useful, however, they found it very difficult to select their intended character. This was because the design added an additional prediction using the Viterbi method based on the distance of the pen cursor to certain keys. The prediction results apparently gave different value than what the test users intended. This made the Cirrin expanded the key of the character next to the intended character. To these test users, this made the Cirrin layout was not stabile. Overall, the function made the selection task difficult and the test users selected wrong characters, especially at the early session (Figure 5.6). The problem with the fisheye style also effected the number of strokes at the early session as shown in Figure 5.7. However, in a long run, the results indicated that the test users of this type of adaptive Cirrin performed better than the Qwerty, the simple Cirrin and the regular adaptive Cirrin.

Table 5.4 shows that in average users only made one character mistake per word. This could be because of adding extra (unnecessary) characters, selecting incorrect completion or selecting the character next to the intended one. The parti-

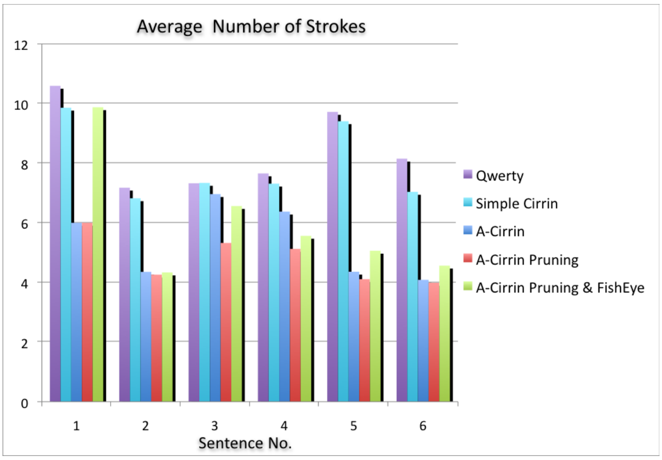


Figure 5.7: *The number of strokes per word (for sentence 1 - 6)*

cipants reported that at the early session, they did not know that there was additional space after a word completion was selected. Therefore, they made mistake by adding extra space after the selection. We also found that the test users of the adaptive Cirrin with pruning suggestion often undid their input after they realized that the previous suggestion was actually their intended word.

Table 5.4: *Correction task*

Text Entry System	**#Incorrect characters in Word**	**#Extra Characters in Word**	**Selecting Neighbor per #Char**	**#Correction Tasks per Word**
Qwerty	0,85 ± 0.02	0.80 ± 0.02	1.03 ± 0.00	0.35 ± 0.04
Simple Cirrin	0.90 ± 0.02	0.59 ± 0.03	0.07 ± 0.00	0.28 ± 0.01
Adaptive Cirrin	0.98 ± 0.10	0.30 ± 0.01	0.07 ± 0.01	0.53 ± 0.03
Adaptive Cirrin with Pruning Suggestion	0.85 ± 0.02	1.00 ± 0.01	0.05 ± 0.01	0.46 ± 0.03
Adaptive Cirrin with Pruning Suggestion and Fisheye Style	1.06 ± 0.18	1.24 ± 0.02	0.08 ± 0.02	1.16 ± 1.23

The experimental results showed that using the Qwerty layout, the participants had a larger problem in selecting the neighbor key than using the Cirrin layout. Additionally, the results also indicated that the test users of the simple Cirrin took their time to search their intended characters before selecting them. Therefore, these users had less correction tasks than other types of Cirrin.

User Satisfaction

In general, the test participants were enthusiastic and found the Cirrin keyboard especially the adaptive Cirrin with pruning suggestion was very useful. They liked the design of the keyboard and its language-based features. 24% of the participants also mentioned that the fisheye style would be useful too if the keyboard display was stabile. Most participants confirmed that they have been familiar with the Qwerty Layout therefore they could perform better using this keyboard than using the Cirrin keyboard. 80% of the participants believed that if they had been familiar with the Cirrin design, they would have performed better using the adaptive Cirrin than on the Qwerty. 63% of the participants liked the adaptive Cirrin better than Qwerty. In addition, despite the input problem, 32% of the participants liked the adaptive Cirrin with the fisheye style better than the simple Cirrin.

Chapter 6

Introduction to Visual Language

In which visual language and its developed interfaces are introduced.

A word completion is generally known for improving user input performance. Instead of inserting characters one by one until the completion suggests the intended word, an icon that represents an object, an action, or a relation, can be selected to replace the word, the phrase or even the entire sentence. Creating messages using such a modality is assumed to result in faster communication. This brings the idea of a new interaction paradigm using visual symbols, such as icons, for representing concepts or ideas to encounter limited options of user interaction provided by handheld devices. As icons offer a potential bridge across language barriers (Perlovsky, 1999), the user interaction on an interface using this type of languages is particularly suitable for a fast interaction in language-independent contexts.

6.1 An Icon as a Representation of a Concept

In contrast to current worldview, which valorizes literacy to the exclusion of other forms of expression/knowing, Singhal & Rattine-Flaherty (2006) argues that images and sketches represent important tools for communication research and praxis. These types of communication provide an alternative to the privileging of text and writing as a mode of comprehension and expression, especially to communicate about topics that are difficult to speak out against. Skills at interpreting visual symbols play an important part in human learning about the world and understanding of language. Words are also composed by symbols, of course. However, there are nonverbal symbols that can provide essential meanings with their succinct and eloquent illustrations. Humans respond to these symbols as messages, though often without realizing exactly what has caused to reach a certain conclusion. Such symbols are often visual, though they can be auditory or even tactile.

An icon is an image, picture or representation. It can refer to anything we know, such as an object, action, or relation. Icons have already been used for intercommunication in the Middle Ages. Because of the imperfection of speech, ways of communication were created. Using symbols, certain concepts could be translated to a physical form, which would be understandable by more. The oldest known symbols used for communication are cave paintings (Figure 6.1(a)) drawn on walls and ceilings during the prehistoric time ($\pm 32,000$ years ago). Another form is petro-glyphs (Figure 6.1(b)), which is created by removing part of rock surfaces by carving or incising.

Figure 6.1: *Ancient ways of communication: (a) cave paintings and (b) petro glyphs*

Advancing further in time, pictograms (pictograph) were created by illustrations to form "sentences" that communicate stories. This type of "communication" was called pictography, a way of communication trough the use of images. Isotypes developed by Neurath (1882-1945) was the pioneer of this type of communication in media literacy and practical system. It was aimed to directly communicate to the masses important facts about their environment and social circumstances, for example in Figure 6.2.

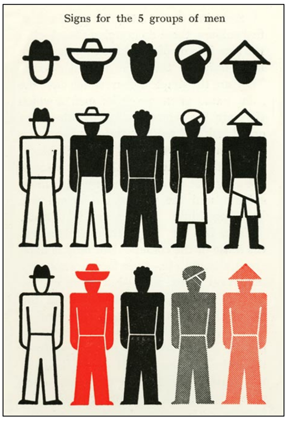


Figure 6.2: *Chart from Neurath's International Picture Language (1936) depicting a newer alternative symbolization of the different human races*

Nowadays, we can find ourselves surrounded by icon-based communication. This ranges from device controls and road sign (Figure 6.3(a)) to such systems assisting speech impairment. Icons form an important part in most GUI-based software as a small graphical representation of a program, resource, state, option or window (Figure 6.3(b)). Typically, these icons are visually different across languages even if they are meant to stand for the same concept. However, since icons are representations of concepts, with which humans are actually interacting, any interaction using icons is expected easy to learn. Once a set of icon-based representations is established, increased usage may lead to more stylized and ultimately abstract representation, as has occurred in the evolution of writing systems, such as the Chinese fonts (Corballis, 2002). Besides having the potential to be a universal language (Perlovsky, 1999), direct icon manipulation allows faster interaction to take place (Kjeldskov & Kolbe, 2002). As pictorial signs, icons can be recognized quickly and committed to memory persistently (Frutiger, 1989). Therefore, they can evoke a readiness to respond for quick exchanges of information and promote quick ensuing actions (Littlejohn, 2001).

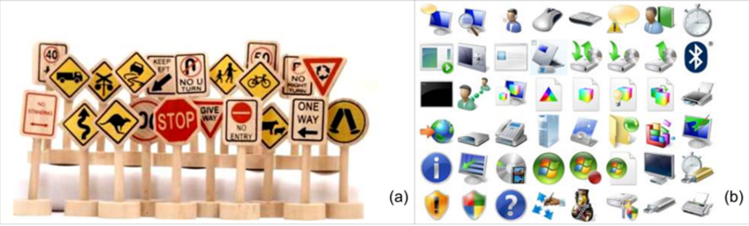


Figure 6.3: *Modern icons: (a) traffic signs and (b) GUI-based computer icons*

6.2 Semiotics

Humans communicate to share facts, feelings, and ideas among each other. This communication involves the use of signs. A sign can be communicated through verbal and non-verbal channels. How can anything come to count as a sign?, how do signs obtain their meaning? and how do they convey them? are questions addressed in the study of signs, *semiotics* (Chandler, 2001). Examples of signs may range from gestures and words to pictures or to nature phenomenon. According to Pierce (1955), a sign is a product of a three-way interaction between the "representamen" (representation that which represents), the "object" (that which is represented) and its mental "interpretant" (the process of interpretation). The representation's goal is to effectively create an interpretation, which matches the object. This interpretation process is called *semiosis*. It has a degree that depends on how close the interpretation in the user's mind is to the object of the icon.

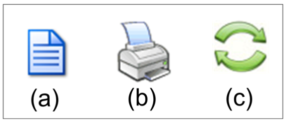


Figure 6.4: *Examples of icons: (a) document icon (iconic sign), (b) print icon (indexical sign) and (c) reload icon (symbolic sign)*

An icon can represent an object through a resemblance (iconic), through causation (the effect of an action or a desired - indexical) or purely through convention (symbolic)(Figure 6.4). The design of iconic and indexical signs can also be conventional. This is because it is impossible to create an exact resemblance in the space available for an icon. For example, the "print" icon is an example of an indexical sign. It does not look exactly like a printer but shares enough of the conventional features that can be recognized as an image of a printer that refers to printing a document. Thus, all icons are symbolic or conventional to some extent.

The interpretation of an icon is created entirely by the user and in the context of that user. Despite this, the semiotic approach does in fact provide guidance to the entire process of icon understanding. In particular, we can analyze the suitability of the icon to the task it is designed to serve. Icons alone are meaningless without a particular context as suggested in the following relationship (Horton, 1994):

$$icon_i + context_j + viewer(orinterpretation)_k \rightarrow meaning_{ijk} \tag{6.1}$$

where the effectiveness of an icon depends on what is represented and how it is represented. Icons almost always exist in the context of other icons. Because of this, the interpretation relation is not quite as free as one might imagine. Based on this theory, Chapter 10 discusses icon design guidelines of a visual language-based interface.

Since signs form human communication means, semiotics can be analyzed along branches of linguistics (Chandler, 2001), such as semantic (the relationship of signs to what they stand for), syntactic (the formal or structure relations between signs), and pragmatics (the relation of signs to interpreters). In addition, semiotics is often employed in the analysis of texts in an attempt to characterize the structure of various signs in text and identify potential meanings. Due to this, communication using signs is often called a meta-language, a language that explains another language.

6.3 Visual Language-based Communication

Visual language refers to the idea that communication occurs through visual symbols as opposed to verbal symbols or words. An essential feature of this language is that sentences are generated based upon a vocabulary of visual symbols. The visual symbols that are referred here, are icons.

Each icon in a visual language-based sentence has unique or multiple meanings (Chang et al., 1994). As a pictorial sign, an icon can communicate more than just its visual contents. This is because a wide range of prior knowledge and experience is brought to bear on the imagery by the viewer, which can be exploited to enrich the meaning of an icon. Since it represents a concept, this symbolic representation of an icon is also counted as a language. An individual icon can be interpreted by its perceivable form (syntax), by the relation between its form and its meaning (semantics), and by its usage (pragmatics) (Chandler, 2001). To cope with the ambiguity of meaning represented by an icon, we can define the meaning by a predominant word or phrase created according to metaphors fitting a given context.

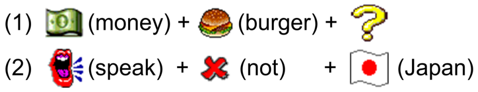


Figure 6.5: *Two visual language-based sentences: (1) "How much money is the hamburger?" and (2) "I don't speak Japanese"*

An visual language-based sentence with more than just one icon can still be represented by the same icons, but it turn out to be difficult to determine the meaning. The only thing that can be automatically derived from the semantics of the icons in a sentence is a fixed word or phrase belonging to these icons (such as "money", "hamburger", "not", "speak" and "Japan" - Figure 6.5). The problem is that individual icons provide only a portion of the semantics of the sentence. The meaning of this type of sentences is derived as a result of the combination of these icons, and cannot be detected without a global semantic analysis of the sentence. It is unlikely that constructing such a sentence with a one to one matching of icons to words would be appropriate. Humans in the other hand are born with the ability to learn language and innate ability to see the structure underlying a string of symbols.

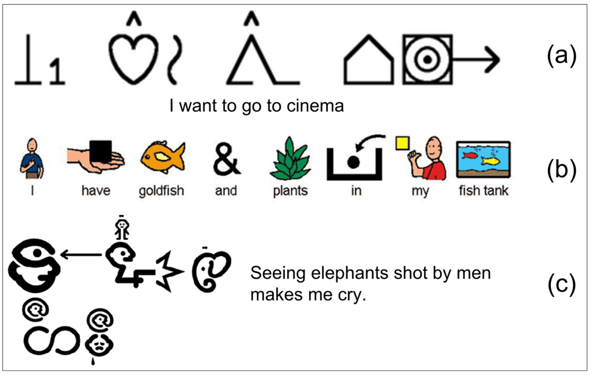


Figure 6.6: *The examples of visual language-based messages on: (a) Blissymbol, (b) Minspeak, and (c) the Elephants Memory*

Recent attempts have been made in developing computerized visual language-based communication systems. The research was mainly aimed at supporting people to communicate with each other despite not sharing a common language. The following are some examples of these communication systems.

Blissymbolics (Semantography). This system contains abstract visual symbols for orthography (Bliss, 1984) (Figure 6.6(a)). It uses combinations of symbols to describe attributes and compose simple symbols for representing new meanings.

CDIcon. This system was designed as a pure person-to-person communication system (Beardon, 1992). A message can be composed as four interconnected screens allowing to specify message types (assertion, question, imperative or negation) and their logical combination (AND, OR, implication, temporal or spatial), primitive act based on the Conceptual Dependency theory (Schank, 1972, 1973), modifiers (color and size), and lexicon as the content of the message.

Minspeak. This system was designed to support individuals with speech impairments (Barrow & Baker, 1982) (Figure 6.6(b)). It maps a set of icons onto a large number of sentences based on the Conceptual Dependency theory . It produces text as output of the interpretation of a visual representation.

The Elephants memory. This system allows the user to build visual messages by combining symbols (logograms) from a predefined vocabulary (Housz, 1994) (Figure 6.6(c)). This language does not have a clear syntax and semantic rules. Formulated visual statements are to be read along non-linear, recursive hyper-textual structures, with the grammatical structure based on position and size parameter.

CAILS (Computer Assisted Iconic Language System). This system uses graphical icons to represent the vocabulary of Basic English developed by Ogden (1930). The icons are placed into simple geometric forms that express grammatical structure (subject, object, ablative, dative, destination and genitive) and tenses. Various arrows are used to express certain types of conjunction (that, if, because, but and with).

Sanyog. This system was designed for disabled people in India (Brar et al., 2004).

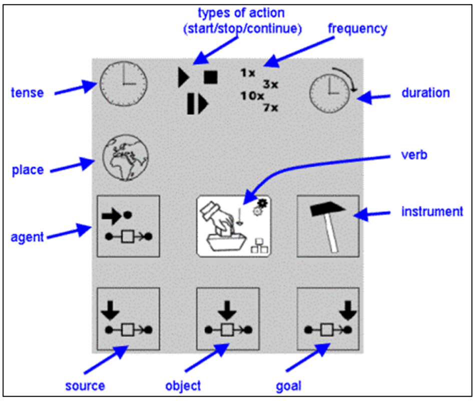


Figure 6.7: *Main screen of the VIL system*

Visual Inter Lingua - VIL. This system was designed based on conclusions drawn from simplified natural languages, such as Pidgins, Creols and Basic English (Leemans, 2001). User needs to select a verb first then all further attributes to be specified in a sentence are determined (Figure 6.7). The system forces the users into strict linguistic structures of thinking and acting when composing sentence.

Except VIL, most systems are hard to learn or language specific. They are either based on too complex linguistic theories or on non-intuitive (or non-self-explanatory) icons.

6.4 The Metamorphic Lingua System

The research into a visual language-based interface described in this thesis started with a project to investigate a new user interaction paradigm on handheld devices. The project was published in Fitrianie (2004). In this paradigm, the icons are used to represent concepts or ideas. They function as communication means, in which their composition can be converted to other modalities, such as text and speech directly. This way leads on developing a visual language-based interface as a communication interface that is independent from any human natural language.

Figure 6.8: *The first prototype of visual language-based interface: a language tool for travelers*

An experimental visual language-based interface that was applied on a language tool for travelers, has been developed. The application was called *Lingua*. It provided handy common utterances for travelers and played as a reference book for the travelers (Figure 6.8). The development of the interface was concentrated on adult travelers within the age 25–50 years old as the target group based on the survey of AvantGo (2003), which has found that 69% of the population of this survey were within the age 25–50 years old, mobile users and using a handheld device.

The first work made preliminary steps in proposing a communication paradigm. Using the visual language-based interface, a user can select a sequence of icons as the realization of his/her concepts (Figure 6.9 - 2004). Human language models in formulating concepts and ideas into a visual language-based messages have been studied. Techniques from Natural Language Processing (NLP) have been adopted to interpret and convert the messages to text and speech. The interface also has an icon prediction that helps the user to have faster interaction in next icon selection. A user test has been performed, which was aimed at an assessment of whether or not the users capable to express their concepts in mind solely using a spatial arrangement of icons.

Although the first results were very encouraging, however, we found that the selected technique restricted to limited visual language-based message creations. For example, creating compound sentences was impossible. Moreover, from the developer's point of view, it was hard to maintain the consistency of the grammar rules if more sentence types and structures were included. In addition, according to Bavelier et al. (1998), the syntax analysis of the visual language does not reduce to classical spoken sentence syntax. There exists a set of two-dimensional (2D) "topic" and "comment" relations of linguistics symbols, in which a comment explains a topic. Conventional textual syntax structures are not considered 2D, since the parser processes them as one-dimensional streams of symbols.

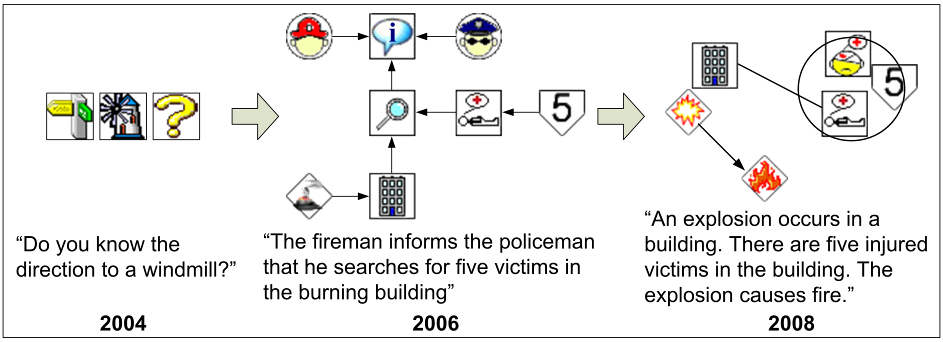

5
5
"Do you know the direction to a windmill?"
2004
"The fireman informs the policeman that he searches for five victims in the burning building"
2006
"An explosion occurs in a building. There are five injured victims in the building. The explosion causes fire."
2008

Figure 6.9: *Examples of visual language-based messages: in a sequential order (developed in 2004), (b) in a 2D configuration (developed in 2006) and (c) using lines, arrows and ellipses in sentence constructions (developed in 2008)*

The second prototype of a visual language-based interface has been developed for reporting observations in the field of crisis management. In this prototype, messages can be constructed using a visual language-based sentences in a 2D way (published in Fitrianie et al. (2006)). The 2D syntax structure was inspired by Deikto (Crawford, 2008) and the case grammar (Fillmore, 1968). Figure 6.9 - 2006 shows the sentence construction of a message. The aim of this work was to improve the visual language grammar that has been proposed by the first prototype.

The latest prototype of a visual language-based interface was aimed at developing such an interface that is able to support a free and natural way to sketch and describe crisis situations (Figure 6.9 - 2008). In this research, we were back to the idea presupposes the use of visual language: "the creation of an image to communicate". A diagram, a map, and a painting are all examples of uses of visual language. Rather than the linear form of symbols (words) used for composing natural language sentences, the elements of visual compositions in the visual language are constructed in a spatial and temporal context. This idea brought us to such an interface that allowed a free arrangement of visual symbols to represent meaning (published in Fitrianie et al. (2008b)). A coherent and context dependent interpretation of the configuration could be constructed by the employment of ontology. In addition, the developed interface was also able to convert the interpretation into crisis scenarios as feedback to the user on his/her input.

The next chapter discusses the two linguistics-based visual language approaches. Then, the native mind of visual language approach is presented in the subsequent chapter.

Chapter 7

Linguistics-based Visual Language

In which the first approach of a visual language-based communication interface is described. Its improvement is also presented.

It is often believed that just as people can verbalize their thinking, they can visualize it. In creating visual language-based messages, research in Fitrianie (2004) shows most people express only important parts or keywords of the their message using two or three icons. This way similar to exchanging message using a telegraph or an SMS. Humans can understand the composed messages because of their innate rules to determine possible shapes of language. In the current research, natural language grammars and methods in NLP have been selected and used to interpret this type of messages and convert them to text or speech. Combining with an inference algorithm and rules, the meanings associated with a visual language-based sentence can be derived from the meanings associated with the individual icons that are forming the sentence.

7.1 Linguistics-based Sentence Structures

Backus Naur Form (BNF) is one of formalisms and notations for writing the syntax that is used to express context-free grammars. The BNF is most commonly used for specifying the syntax of programming languages, command sets and language descriptions. The syntax is described by a set of phrase structure rules and their transformations. It is defined as a set of strings. Each string is a sequence of symbols from a finite set called the terminal symbols. For English, these terminal symbols include words. Non-terminals, such as noun phrase (`NP`), verb phrase (`VP`) and sentence phrase (`S`), are used in the construction of a sentence, but they do not actually appear in the final sentence. Phrases, such as "the post office" and 'my money', are examples of the category `NP` whereas construction with a verb, for example "is closed" form `VP`. Any `NP` can be combined with a `VP` to form a phrase of the category `S`. With the rule: `S := NP VP`, for example, we can explain that "the post office is closed" is a sentence whereas "the post office closed is"' is not. Here ':=' can be interpreted as "produces".

The phrase rule specifies precedence relations (for example, `NP` precedes `VP`) and hierarchical relations (for example, S immediately dominates NP and VP). A visual language usually concerns with the hierarchical composition. Fillmore's case grammar (Fillmore, 1968) is an example of linguistic analysis theories that follows the hierarchical sentence composition. This theory analyzes the surface syntactic structure of sentences by studying the combination of deep cases. The cases (semantic roles) are the notions of conceptualizations and concepts that are units of meaning loosely corresponding to the grammatical units of clauses and words (such as noun, verb and adjective) in a sentence construction.

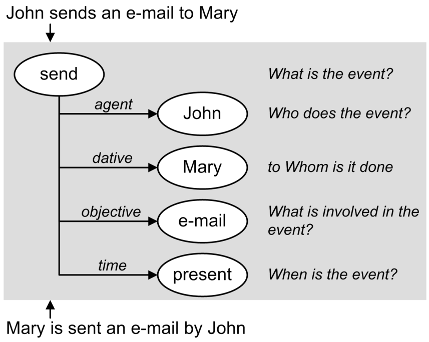


Figure 7.1: *An example of a sentence structure in a case grammar*

In the case grammar, a sentence in its basic structure consists of a verb and one or more noun phrases, each associated with the verb in a particular case relationship. A compound of instances of a single case can be formed through noun phrase conjunction. Figure 7.1 shows two sentences that have different surface structures, but the deep structures are the same. In this grammar, the main verb in the proposition is the focus around which the other phrases revolve and the auxiliary verbs contain much of the information about modality. The case notions make up a set of universal concepts, which identify certain types of judgments human beings are capable of making about the events that are going on around them. Some example cases are agentive (actor of the action identified by the verb), instrumental (the object or force involved in the action state identified by the verb), dative (the recipient of the state or action identified by the verb), locative (the location or spatial orientation of the state or action identified by the verb) and objective (the thing being acted upon).

VerbNet is a verb lexicon with syntactic and semantic information for English verbs (Schuler, 2005). It refers to verb classes defined by Levelt (1989) for constructing the lexical entries. The classification is based on the ability of a verb to occur in pairs of syntactic alternations, which preserve the underlying semantics. The verb classes are hierarchically organized. Each node is characterized extensionally by its set of verbs, and intentionally by syntactic and semantic information about the class and a list of typical verb arguments. The argument list consists of thematic labels and possible selection restrictions on the arguments expressed using binary predicates. The syntactic information in the entry of each verb maps the list of thematic arguments to the deep-syntactic arguments of that verb. The semantic predicates list the participants during various events described by the syntactic frame. The syntax describes constructions, such as transitive, intransitive, prepositional phrase complement, resultative and a large set of Levin's alternations. A syntactic frame consists of the verb itself, the thematic roles in their preferred argument positions around the verb, and other lexical items that may be required for a particular construction or alternation. For example, the syntax `Agent V Patient` for "John hits the ball", `Agent V at Patient` for "John hits at the window" and `Agent V Patient[+plural] together` for "John hits the sticks together". The semantic information for the verbs is expressed as a conjunction of semantic predicates, such as motion, contact and transfer-into. The lexicon has been mapped to WordNet (Fellbaum, 1998).

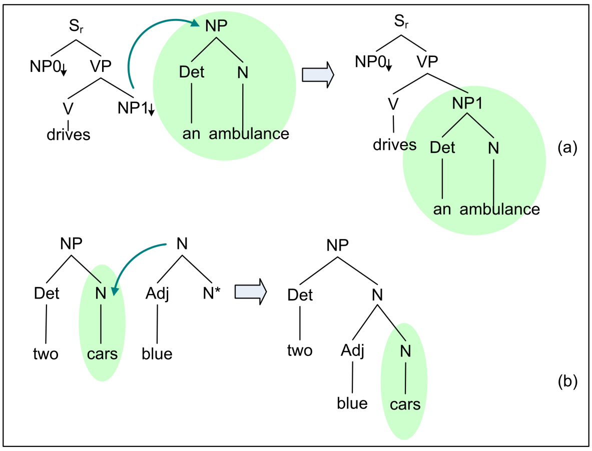
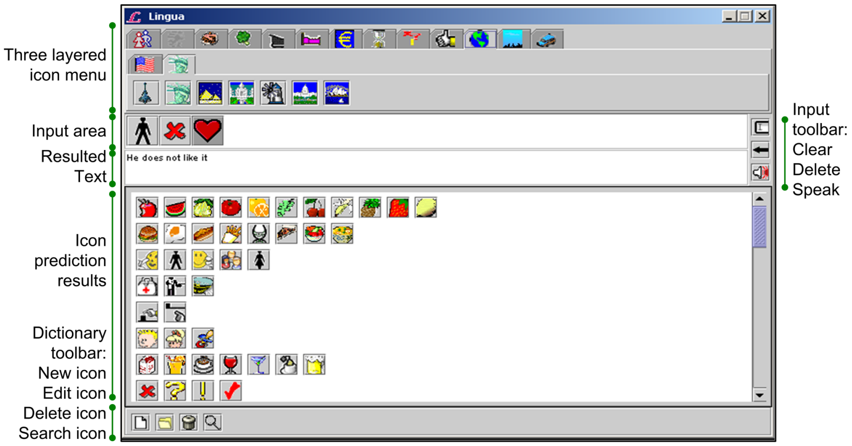


Figure 7.2: *Operation on TAG: (a) substitution and (b) adjunction*

To extend VerbNet's syntactic coverage, Ryant & Kipper (2004) has incorporated the coverage of Xtag by accounting for the possible transformations of each declarative frame in VerbNet. The Lexicalized TAG (Xtag) is a set of lexicalized elementary trees (XTAG-Project, 2001; Doran et al., 2000; Joshi, 2001), which is developed based on Tree Adjoining Grammars (TAGs) (Joshi & Schabes, 1997). To derive syntactic structures for sentences, the trees can be combined, through the operations of tree substitution and tree adjunction (Figure 7.2). The extension of the VerbNet is done by mapping its syntactic frames onto elementary trees of TAG families. For any verb in VerbNet, each thematic role can be mapped to an indexed node in the basic syntactic tree and the selectional restrictions on the VerbNet's thematic roles to features on the nodes.

7.2 Sequential Arrangements of Icons

Figure 7.3 shows the interface of Lingua. A user can select a sequence of icons to create a visual language-based sentence as a realization of his/her concepts or ideas in mind. The visual language-based interface is able to interpret the sequence and convert it into a human natural language text and speech.


Lingua
Three layered icon menu
Input area
Resulted Text
Icon prediction results
Dictionary toolbar: New icon Edit icon Delete icon Search icon
Input toolbar: Clear Delete Speak

He does not like it

Figure 7.3: *The main screen of the Lingua system*

The interface was developed to cope with the screen size of a handheld device (mostly 640x480 pixels). The current version has five hundred icons in its vocabulary divided into thirteen concepts. Some of these icons were collected from the Internet. Therefore, they have differences in quality and resolution.

A user can select icons from the icon menu, the prediction list or using the search window. The icons are displayed in groups based on their concept. The icon menu displays the icon vocabulary in three levels. The top level presents the concept icon and incorporates compound icons. Querying any concept will initiate a move to the second level, at which can contain sub-concept icons or base icons. A further level contains only base icons. On each selection, the interface shows a distinctive appearance for icons that cannot be selected any more based on English grammatical rules.

The user can compose a message in the input area. Selected icons in the input area can be deleted using the option in the input toolbar. Each time an icon is inserted or deleted, a textual conversion of the inputted icons so far will be updated directly. Using the input toolbar, the user can also activate the TTS synthesizer.

7.2.1 BNF-based Grammar

Fitrianie (2004) reported two studies into developing the grammar rules for messages using a sequential arrangement of visual symbols using BNF and English grammars: (1) a workshop for acquiring knowledge on formulating visual language-based messages from natural language sentences and (2) an analysis of a large number of corpora of visual language-based messages as a comparison. Figure 7.4 shows some examples of the corpus that are expressed by sequences of icons.

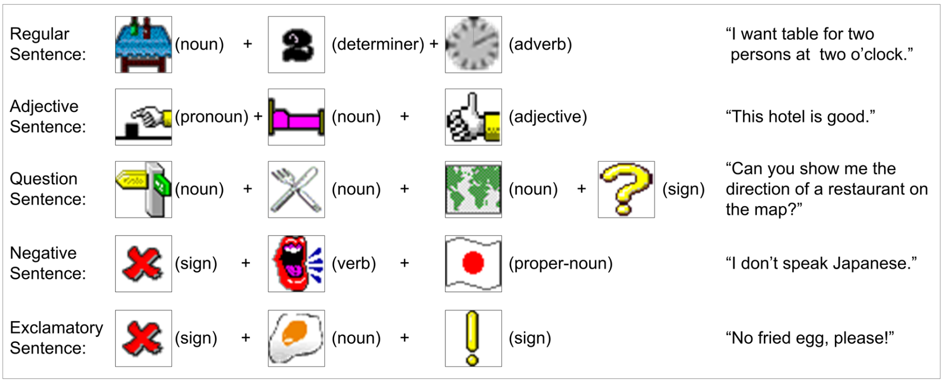

Regular Sentence:
(noun) + (determiner) + (adverb)
"I want table for two persons at two o'clock."
Adjective Sentence:
(pronoun) + (noun) + (adjective)
"This hotel is good."
Question Sentence:
(noun) + (noun) + (noun) + (sign)
"Can you show me the direction of a restaurant on the map?"
Negative Sentence:
(sign) + (verb) + (proper-noun)
"I don't speak Japanese."
Exclamatory Sentence:
(sign) + (noun) + (sign)
"No fried egg, please!"

Figure 7.4: *Examples of sentences in a BNF-based grammar of visual language*

The first step in defining a grammar was to define terminal symbols, such as lexicon or list of allowable vocabulary icons. The icons are grouped into the categories or POSs familiar to the dictionary of the users, such as nouns, pronouns, proper-nouns, verbs, adjectives, adverbs, prepositions and quantifiers. The grouping depends on what an icon represents. For example, the icons that represent "table", "hotel", "direction" and "map" are nouns, the icon that represents "speak" is a verb, the icon that represents "Japan" is a proper-noun, and the icons that represent "not", "!" and "?" are signs.

The next step was to combine the icons into phrases as non-terminal symbols, such as S, NP, VP, and Prepositional Phrase (PP). As examples using the negation sentence in Figure 7.4, the grammar rules are:

```
NP := pronoun | proper-noun | pronoun NP
VP := sign verb NP
S :=  NP VP
```

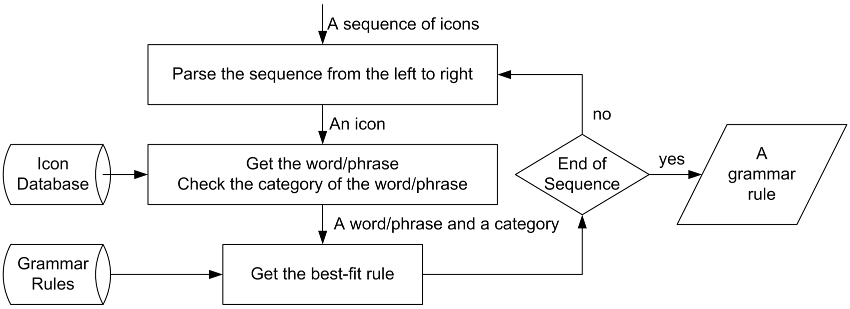
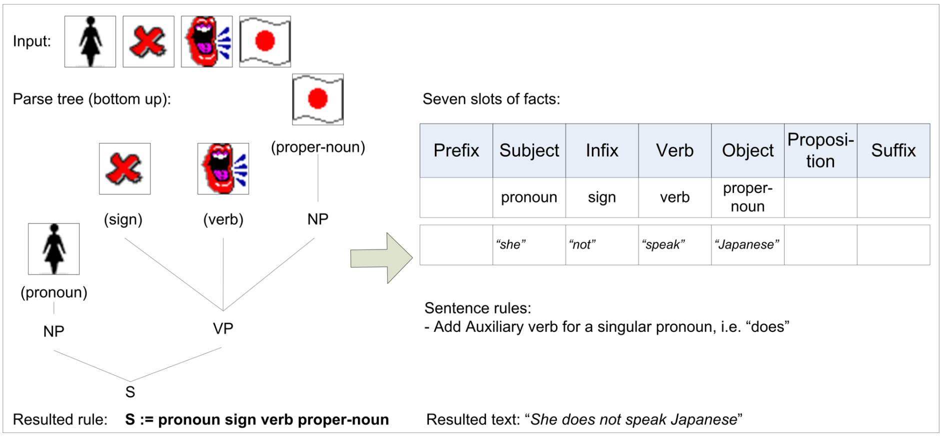


Figure 7.5: *The flowchart of the syntax analysis*

The final step was to develop a parser that takes a visual language-based sentence as input and extracts a meaning for the sentence as output using the developed grammar rules. This parser parses the sentence from the left to right into a sequence of tokens. A token in this case is an icon. Figure 7.5 shows the flowchart of the syntax analysis process. The icon database defines the terminal symbol of each icon. The parser takes every icon from the sequence and matches it with every terminal symbol according to the grammar rules. The algorithm stops if an S is formed and a matched grammar rule is selected. If there is not any rule, then this parser yields a syntax error.


Input:
Parse tree (bottom up):
(sign)
(verb)
(proper-noun)
NP
(pronoun)
NP
VP
S
Seven slots of facts:
Resulted rule: S := pronoun sign verb proper-noun
Resulted text: "She does not speak Japanese"

Prefix	Subject	Infix	Verb	Object	Proposi-tion	Suffix
	pronoun	sign	verb	proper-noun		
	"she"	*"not"*	*"speak"*	*"Japanese"*		

Sentence rules:
- Add Auxiliary verb for a singular pronoun, i.e. "does"

Figure 7.6: *An example of the syntax analysis for "she does not speak Japanese". The parser translates each icon to a terminal symbol in the icon database and matches it with the grammar rules, transforms the selected grammar rule into seven slots, selects required sentence rules, and constructs a complete sentence.*

If the inputted visual language-based sentence is syntactically correct, the parser creates seven slots: (1) ***prefix slot***, a container for adding a question word, (2) ***subject slot***, a container for the subject phrase of the sentence, (3) ***infix slot***, a container for a to be, an auxiliary and a negation, (4) ***verb slot***, a container for the verb phrase of the sentence, (5) ***object slot***, a container for the object phrase of the sentence, (6) ***proposition slot***, a container for the proposition phrase of the sentence. and (7) ***suffix slot***, a container for a question sign and an exclamation mark. Figure 7.6 displays the input parsing into a sequence of seven slots. The position of the slots may not in a sequential order. It depends on the type of a sentence. For example, for a question sentence, the infix slot may be located between the prefix and the subject slot. A slot may be empty and may contain more than one word. To complete the sentence, some extra rules are included in the parser. Table 7.1 shows the sentence rules specifically for creating a simple present tense sentence.

Table 7.1: *A list of sentence rules in the current version of the BNF-based grammar parser*

No.	Rules
1.	Capitalize the first character
2.	Add an article ("a", "an" or "the")
3.	Add a to be ("am", "are" or "is")
4.	Add a question word (for instance "what", "where", "how much", "how many")
5.	Add an auxiliary ("do" or "does")
6.	Change the pronoun's format as an object (for instance "me", "you", "him", "her")
7.	Change the pronoun's format to possessiveness (for instance "my", "your", "his")
8.	Change the verb's format (for example "go" is changed into "goes")
9.	Add preposition (for instance "to", "on", "at", "with", "for")
10.	Change the plural noun's format (for example "bread" into "breads")
11.	Add a pronoun to subject slot if empty
12.	Add object slot if necessary (for example after verb-transitive)

7.2.2 System Design

The first prototype of the visual language-based interface was designed for multilingual use. To be able to "speak" other languages, the interface has the TTS synthesizer and the language translator. The output of the interface is the interpretation of a sequence of icons into a meta or an universal language, such as a set of concepts in English ontology. The Lingua system consists of into three main parts: (1) the Lingua Application, (2) the TTS Synthesizer, and (3) the Language Packet (Figure 7.7). The system was tested in the environment of the *Sharp Zaurus SL-C760 Personal Mobile Tool*. The TTS synthesizer and the language packet were external software and produced by third-party. For experiment purposes, Carnegie Mellon University's `Flite` was used for the TTS synthesizer. The current version of the system does not connect any language translator. Therefore, the resulted text and speech are still in English.

From the technical point of view, it was difficult to develop the grammar and the sentence rules. Firstly, we have to examine a large number of corpus visual language-based messages. It is necessary to study various sentence structures that include different situations and different structure of word classes. Secondly, it was difficult to maintain the consistency of the grammar and the sentence rules when a large number of sentence structures should be included.

7.2.3 User Testing

We have performed a user test to assess whether or not users were capable to express their concepts in mind solely using a spatial arrangement of icons. This test also addressed usability issues on interacting with the interface.

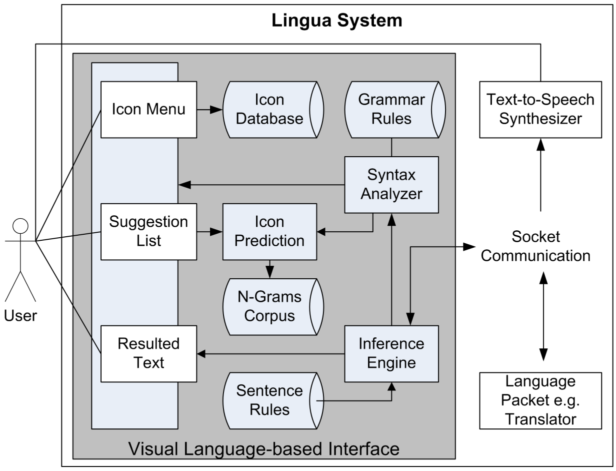


Figure 7.7: *The architecture of the Lingua system*

Methodology

The experiment was performed using Thinking Aloud method (Boren & Ramey, 2000). This technique encourages a natural conversation situation between the experimenter and the participant. In this protocol, the participants are asked to verbalize their thoughts as they take part in the experiment. Eight people took part in the test. They were selected in the range age of 25-50 years old and to include different nationalities.

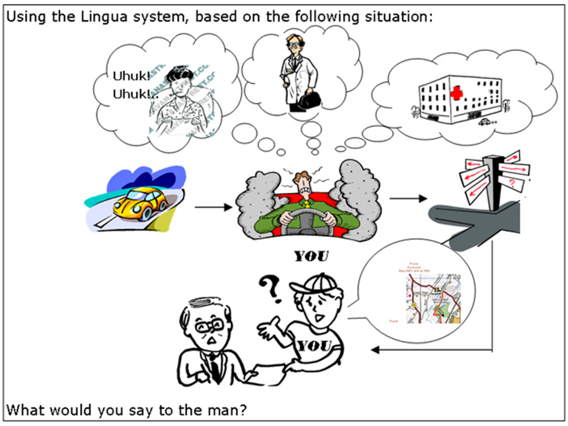

Using the Lingua system, based on the following situation:
Uhuk!
Uhuk!
YOU
YOU
?
What would you say to the man?

Figure 7.8: *An example of the experiment tasks*

Each participant had five tasks. The tasks were created using real situation for travelers. They were cartoon-like stories, which illustrated the situation when the participants used the interface to communicate with others (Figure 7.8). For this experiment, the visual language-based interface had 415 icons classified in 13 groups and the icon prediction contained 513 bigrams and 327 trigrams. There were not incorrect answers, except if the participant did not answer the question at all. All activities were recorded on a tape recorder and all user interactions on the interface were logged, to be analyzed afterward.

The measurement of the sense of being lost in hypermedia (Smith, 1996) was adapted to measure disorientation of being lost in a visual language-based interface. Since all information was displayed only by icons, such an interface is considered that might give cognitive overload and disorientation to its users. The problem referred to is "users cannot find what they are looking for". For a perfect search, the *lostness* rating should have been as low as possible.

Result Analysis

A measure for understanding the concept of visual language-based messaging as applied in the developed interface was whether or not a defined task was accomplished. To deal with visual language-based messaging was not a problem for any participant. They accomplished their tasks with relevant answers. Only 20% of the total tasks, we found that the participants tried to find icons for each word. This usually occurred in the early session when our participants tried to familiarize with the interface. In 85% of the total tasks, the users knew that they would not find an icon for question words, such as "where" and "can you show me ...", or verbs, such as "would like to ..." or "want to". They selected few icons that represent important or keywords of a sentence, and let the interface interprets their selections into a complete sentence. The test users also could sense when a verb was important in a visual language-based message, otherwise they represented it using only a sequence of noun icons.

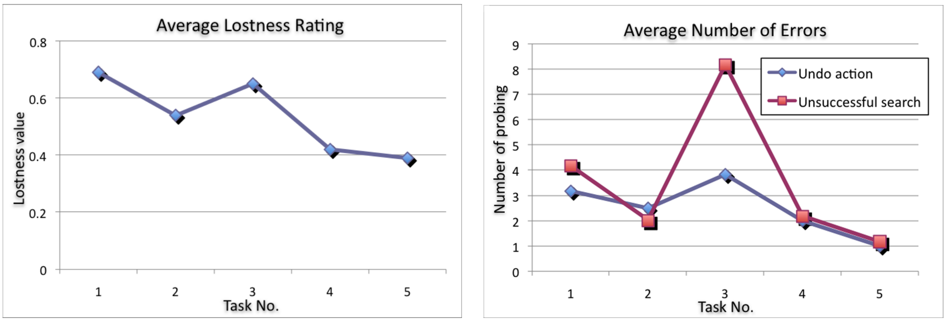
The high value of the lostness rating (Figure 7.9(a)) might indicate that the cognitive process of the test users was influenced by:

- Searching process of a relevant concept to represent the message. It appeared that the test users tended to keep searching the intended icon before they rethink another concept to represent the message. This unsuccessful searching occurred on average 8.17 ± 0.23 times for task no. 3 (Figure 7.9(b)).

- Fail to recognize the provided icon(s).

- Error of the sentence conversion. For example, for a sequence of icons "money + question sign" would result "How much is money?" instead "How much money is it?". This makes the test users uncertain of their own choice and tended to rethink another solution.

- Curiosity to know the representation of an icon. This problem created many undo actions occurred in early tasks (Figure 7.9(b)).



(a) (b)

Figure 7.9: *(a) The average lostness rating for five sequence tasks and (b) the average number of undo actions and unsuccessful searching within five sequences of tasks*

In general, the participants were enthusiastic and confirmed that the developed interface had met their expectation. They liked the idea of representing a sentence only by selecting two or three icons at most. The following are suggestions for the improvement of the visual language-based interface: (a) designing icons that are suitable also people with sight problems, (b) displaying a visual discrimination for the heading icons and the base icons, (c) displaying only the highest probability next possible icons, and (d) designing concept icons that can represent the group. Apart from these problems, the decreasing lostness rating could be viewed in terms of improvement of user performances. The test users had taken benefits provided by the interface, such as the icon prediction and the search engine, in message creations. It was concluded that the test users only needed a small period of time to adapt to the interface.

7.3 A Two-Dimensional Visual Language Grammar

The second prototype of a visual language-based interface was aimed at improving the language model of the first proposal, using a 2D grammar. We combined the concepts from the case grammar (Fillmore, 1968) and Deikto (Crawford, 2008). Deikto is a language between players in an interactive game. A sentence, in Deikto (Figure 7.10), is constructed by an acyclic-graph of concepts, where the predecessor explains the successor concept, such as a word/phrase or a clause. To help the players, the game provides hints of what concept(s) can be filled in given a certain word class. Deikto follows a rigid grammar by assigning to each verb the parts of the sentence in its dictionary definition.

The second visual language-based interface was developed for reporting observations of crisis situations in the field of crisis management (Figure 7.11). Different sets of icons have been developed (see Chapter 10). A user can arrange an acyclic graph of icons, as a realization of his/her concepts or ideas. The interface supports a fast inter-

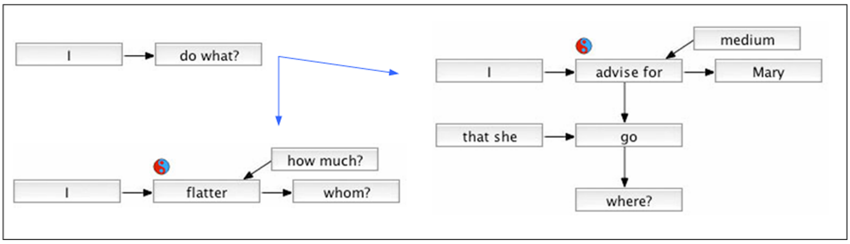
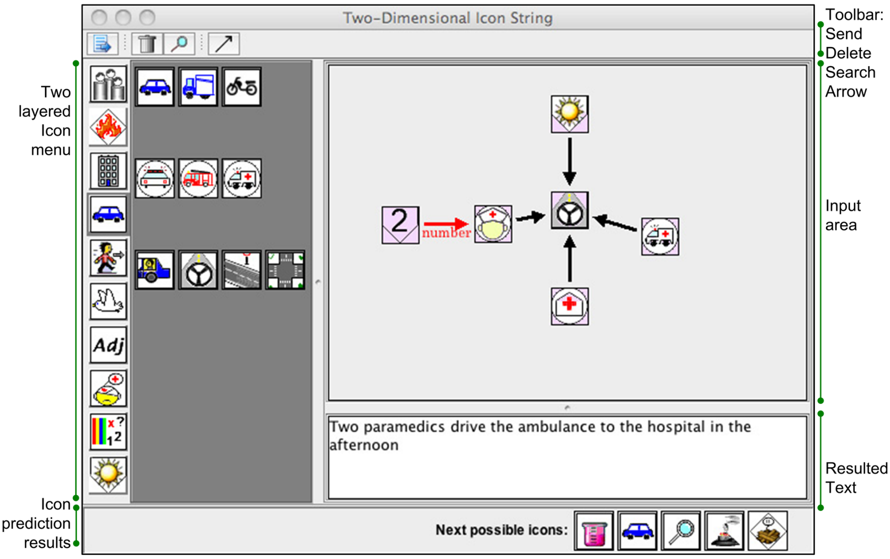


Figure 7.10: *Two examples of Deikto sentences*

action by converting the message into natural language directly. The design of the interface still follows the same concepts of the Lingua system. However, instead of using three layered icon menu, the interface uses two layers. The top level still presents the concept icon and incorporates compound icons. The second layer contains groups of base icons. Distinctive appearances for icons that cannot be selected anymore based on the grammar rules are also shown on each selection. The interface also provides an icon prediction and a search engine. In addition, this interface provides an option to send the observation in the toolbar.



Two paramedics drive the ambulance to the hospital in the afternoon

Figure 7.11: *The interface of 2D visual language-based communication*

To help the user, the attributes of each selected icon on a 2D sentence are displayed in forms of icons (Figure 7.12(a)), such as "who is the agent/experiencer?", "where is it happening?", "when does it occur?", "how many are they?", and so on.

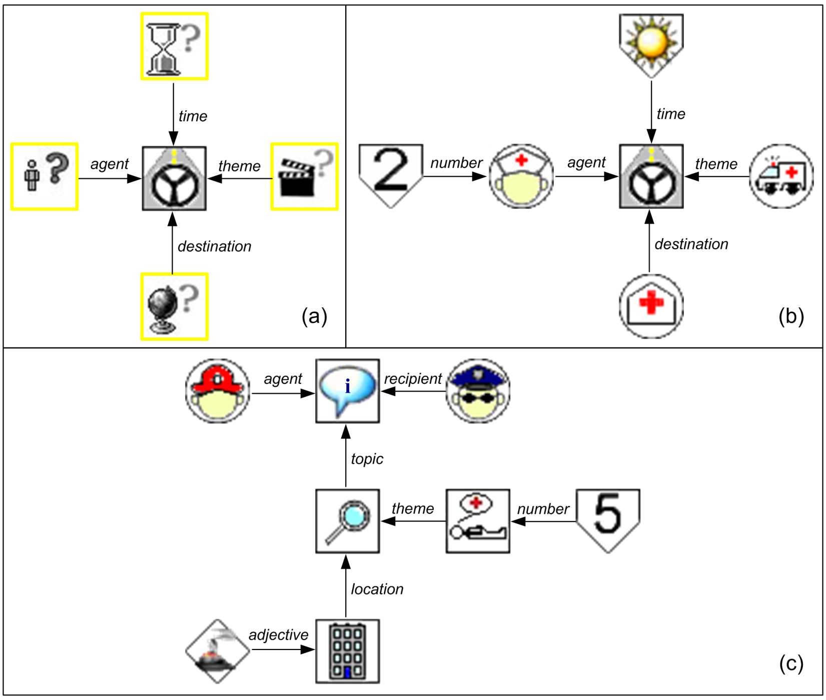


Figure 7.12: *Examples of 2D sentences: (a) hints for the verb "drive", b) a simple sentence: "Two paramedics drive an ambulance to the hospital in the afternoon" and (c) a compound sentence: "The firefighter informs the police that he will search five victims in the burning building"*

Based on the case grammar, these attributes represent the semantic roles of the selected icon. They are displayed with different color border (in this case is yellow). As an icon is deselected, such hints will disappear to reduce the graphic complexity. A hint icon can be selected and replaced by an icon that is grammatically correct to form a sentence. The approach gives a freedom to users to fill in the parts of a sentence, but at the same time it can restrict the choices of icons that lead to a meaningful sentence. A user may attach and connect icons manually to a specific icon in the sentence.

7.3.1 Lexicalized TAG by Visual Symbols

Figure 7.12(b) shows an example of a 2D sentence that is constructed in an acyclic-graph of visual symbols, such as icons. An icon can be connected by an arrow to another icon, in which the former explains the latter icon. Each icon represents a concept or an idea. The sentence may be constructed from any part, however as soon as a verb is selected, the structure of the sentence will be determined. The developed grammar allows a compound sentence construction, which can be done by connecting another verb to the main verb of the sentence (Figure 7.12(c)).

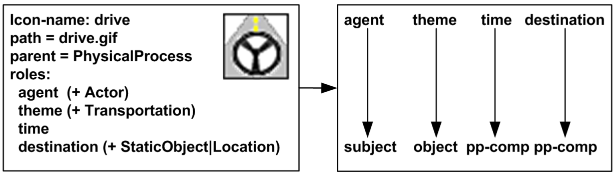


Figure 7.13: *Schematic view of the two components of a verb lexical definition: semantic types and linking to syntactic arguments*

For each icon that represents verbs, a case is defined following the frame syntactic analysis used for generating the VerbNet (Schuler, 2005). For example, the case of the "drive" verb contains agent, theme, location, time (Figure 7.13). A verb is defined as a lexeme that has one or more sense definitions. The definitions consist of a semantic type with associated thematic roles and semantic features, and a link between the thematic roles and syntactic arguments. The definition also defines required and optional roles. A case for other POS icons is defined using properties that may be required for a particular construction or alternation of these icons. For example, the case of the "paramedic" icon may contain number (frequency), living condition, emotion, location, size and so on. These cases are used as a reference to build the hints in a message creation.

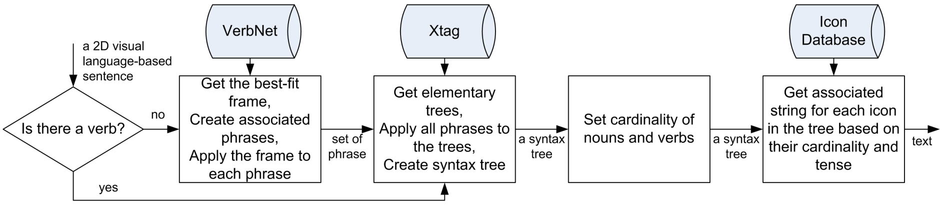

VerbNet
Xtag
Icon Database
a 2D visual language-based sentence
Is there a verb?
no
yes
Get the best-fit frame, Create associated phrases, Apply the frame to each phrase
set of phrase
Get elementary trees, Apply all phrases to the trees, Create syntax tree
a syntax tree
Set cardinality of nouns and verbs
a syntax tree
Get associated string for each icon in the tree based on their cardinality and tense
text

Figure 7.14: *The flowchart of the syntax analysis for 2D visual language-based sentences*

Figure 7.14 shows the flowchart of the syntax analysis of a 2D sentence. A parser processes a 2D stream of icons. It maps their case into the syntax frame of the VerbNet-based vocabulary. The mapping of the Xtag into the VerbNet syntax frames based on (Ryant & Kipper, 2004) has provided a pre-existing rules or structures for translating 2D sentences. The combination naturally and elegantly can capture the interaction between the graph structure of the visual sentences and the tree structure of Xtag syntax and the inferential interactions between multiple possible meaning generated by these sentences. Figure 7.15(a) shows the example of the iconized TAG vocabulary. Presumably, the transformation of the syntactic frames is recoverable by mapping the 2D sentence onto elementary trees of the TAG families. This mapping increases the robustness of the conversion of 2D sentences to natural language text/speech.

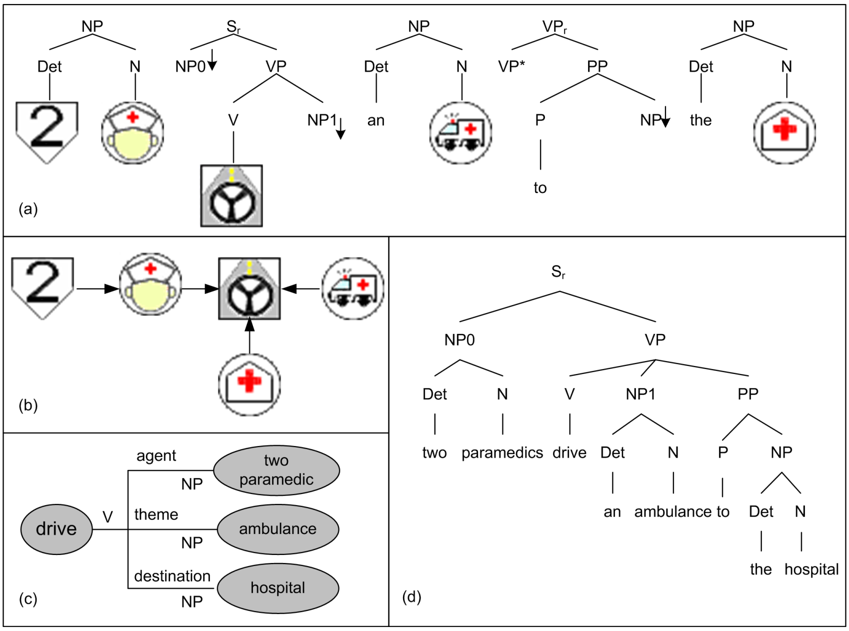


Figure 7.15: *The conversion to text: (a) examples of the iconized TAG elementary trees, (b) a 2D sentence: "Two paramedics drive an ambulance to the hospital", (c) mapping the thematic roles to the basic syntactic tree defined by the case of the verb "drive", and (d) a parse tree as the results of mapping the basic syntactic tree to the TAG trees*

The parser exploits the icon database for the syntactic arguments of every icon in the sentence. Figure 7.15 shows an example of parsing a 2D sentence. The semantics of the sentence are specified in two ways. First, the meaning of a TAG tree is just the conjunction of the meanings of the elementary trees used to derive it, once appropriate elements in a case are filled in. Secondly, the VerbNet's structure provides an explicitly constructed verb lexicon with syntax and semantics. In this way, the syntax analysis and natural language construction can be done simultaneously.

7.3.2 System Design

Figure 7.16 shows the class diagram of the 2D visual language-based interface. The interaction with users is handled by the `FRigid2D` class, which includes handling the message creation, the display of the hints and the display of the icon prediction results. The `2DGrammar` class interprets a 2D icon string into text. It handles the syntax structure of a set of 2D sentences in the `2DAlinea` class. The information of an icon in the string is stored in the `2DWord` class. This includes the cases of the icon vocabulary. A `2DPhrase` can contain one or more `2DPhrase` objects or `2DWord` objects. A list of interconnected 2DPhrase objects forms the syntax tree of a 2DSentence object.

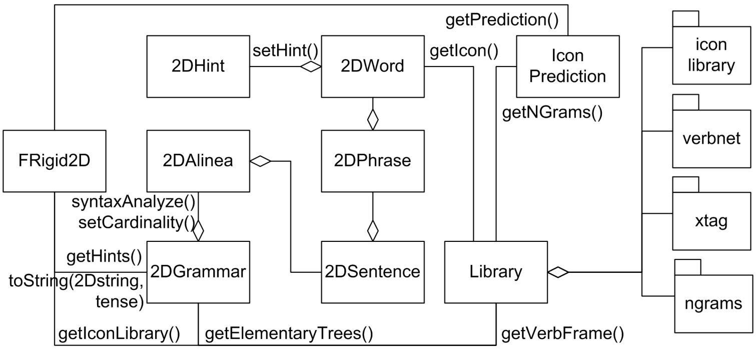


Figure 7.16: *The class diagram of the 2D visual language-based interface*

To interpret the 2D icon string, the 2DGrammar class uses the classes in xtag and verbnet modules. It also exploits the iconlibrary module, which contains the icon database, to convert the string into text. The conversion method needs two parameters: (1) the cardinality of all elements in the syntax tree and (2) the tense of the grammar. The current implementation still focuses on the simple present tense. The development of the interface took the recommendations formed from the results of the user testing of the Lingua system prototype, into account. This included recommendations on better icon design, a simple icon menu, a simple list of the icon prediction results and providing tool tips on the icon vocabulary.

7.3.3 User Testing

We have performed a user testing similar to the sequential visual language-based interface. The test was aimed at assessing whether or not users were still capable to express their concepts in mind using visual symbols provided in a 2D way. The test also addressed the usability issues on the second prototype interface.

Methodology

Eight people took part in the test. The tasks were created using images of real crisis situations (Figure 7.17). The participants were asked to report what they might experience, by creating 2D visual language-based sentences on the interface. While performing the task, they were also asked to think aloud. For this experiment, the icon database of the interface contained 110 icons classified into 10 groups. The icon prediction contained 97 bigrams and 73 trigrams. The interface used the VerbNet v.3.1, which contained 274 first-level verb classes and 5257 verb senses and used 23 thematic roles. The Xtag grammar was developed based on the technical report of XTAG-Project (2001). Only 16 of the 57 elementary trees were mapped directly to the VerbNet structure. 22 Xtag trees families were used to construct the final syntax trees, which dealt with small clauses. Other 19 trees were not used because they dealt with idiomatic expressions and other various classes, which were out of the scope of the developed visual language.

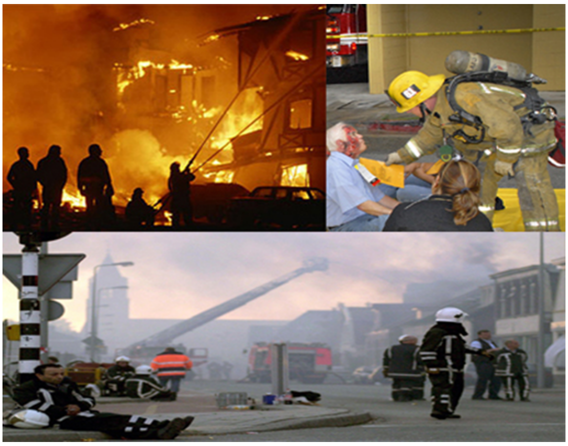


Figure 7.17: *An example of the test cases*

Result Analysis

The experimental results showed that the test users were able to compose 2D visual language-based messages to express their concepts and ideas in mind. As we expected, the participants had to learn to use the hints in the beginning of the test. After a while, the results indicate that the hints helped the users to compose a complete report. There were not any significant new results comparing to the experiment for the first prototype. The users still needed some time to find another concept when they could not find a relevant one from the provided visual symbols to represent their message. They also needed adaptation time to recognize some icons. The only remark was on the icon prediction, which still could not give relevant suggestions due to the lack of the bigram and trigram corpus. Apart from these problems, the experimental results also showed that the test users needed a small period of time to learn to create 2D sentences and to utilize the interface.

From the technical point of view, the developer did not have to work hard on maintaining the consistency of the grammar and the sentence rules anymore. Both the VerbNet and the Xtag library have provided the means. Moreover, this approach allows the creations of rich message structures. The proposed visual language can be applied in different domain in a straightforward manner. For this purpose, the developer only needs to develop and customize the icon database into the domain in focus.

Chapter 8
Native Mind Visual Language

In which the final proposal of a visual language-based interface is described.

In previous work, a translation module that uses a dedicated grammar for interpreting and converting visual messages into natural language text and speech has been developed by adopting approaches of NLP. Hints, navigation and visual cues, icon prediction, and search engine are provided to help users with message creation. However, the grammar still forces the users to follow linguistic lines behind sentence constructions. This makes the previous solutions somewhat rigid and requires the users to learn a "new" grammar. A free and natural way of creating a spatial arrangement of graphics symbols (such as icons, lines, arrows and shapes) to represent concepts or ideas is proposed.

8.1 Thinking in Pictures

Visual language is known as the basic literacy in the thought processes and the foundations for reading and writing. Visual symbols are nonverbal representations that precede verbal symbols (Sinatra, 1986). Pictures or illustrations are analogs of experiences and are only one step removed from actual events. These visual representations may be able to capture and communicate the concrete experiences in various ways (Cohn, 2003). Parallel with verbal language, visual language is used to record and communicate these world phenomena (Figure 8.1).

Children learn by seeing and recognizing their environment (Berger, 1972). Silverman (2002) has reported that 30% of children are thinking mainly in form of visual (spatial) and 45% uses both visual (spatial) and words. Visual Thinking is the ability to turn information into pictures, graphics, or forms that help communicate the information (Wileman, 1993). This ability is associated strongly with the ability to read, interpret and understand information presented in images (Heinich et al., 1999). This process is refined over time, but remains basically the same into adulthood.

The sentence structure of a visual language is different from a sentence in spoken language. The spoken language is composed by a linear ordering of words, while a visual language has a simultaneous structure with a parallel temporal and spatial configuration (Bavelier et al., 1998; Lester, 2006). The elements in a visual symbol represent concepts in a spatial context. Its structural units include line, shape, direction, color, motion, texture, pattern, orientation, scale, angle, space and proportion. Human perception creates a continuous judgment of visual element relationships, which

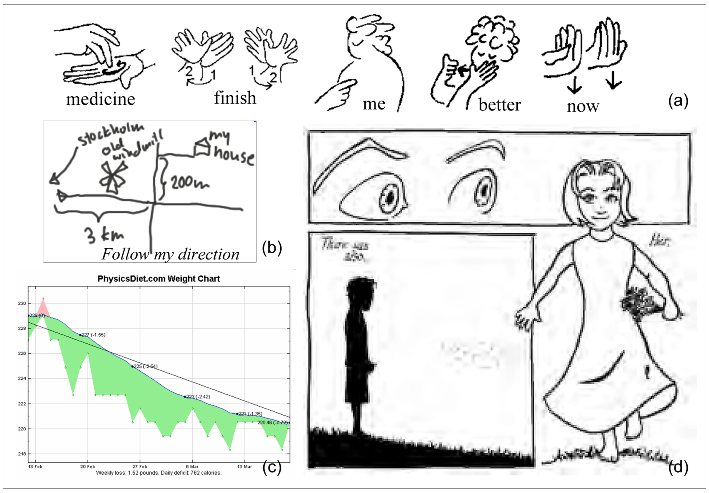


Figure 8.1: *Examples of visual language: (a) sign language, (b) a sketching and (c) a diagram, and (d) a comic illustration*

includes making categories of forms to classify and relate images and shapes in the world (Arnheim, 1969; Chin et al., 2006).

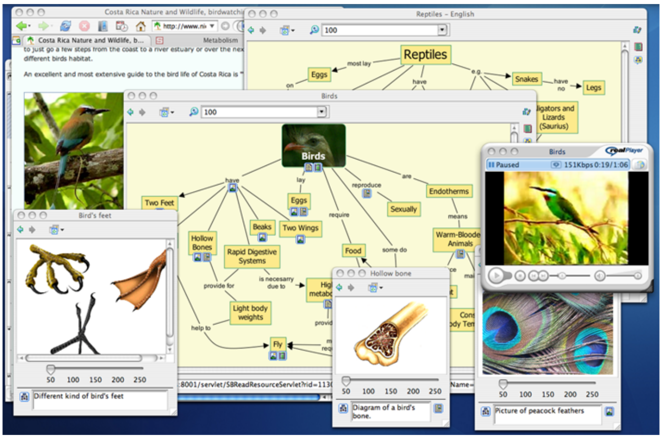
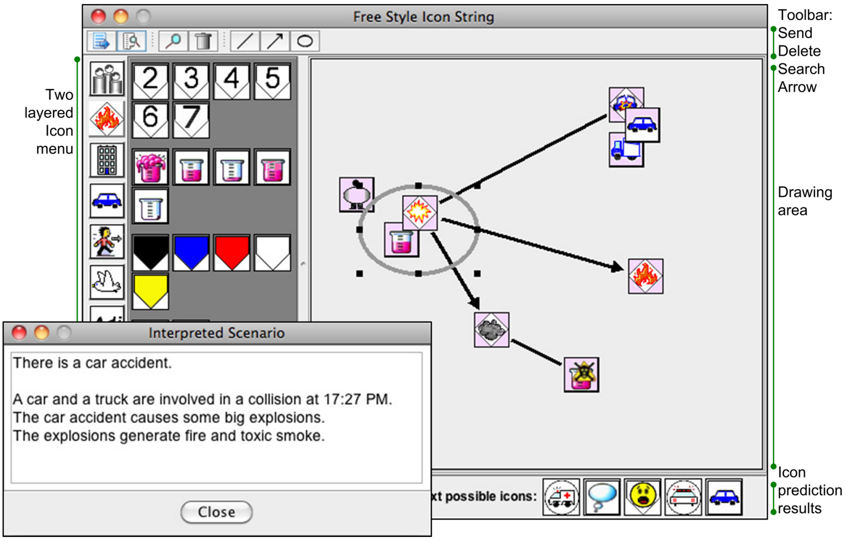


Figure 8.2: *A concept-map about birds constructed by a high-school student. Icons under the concepts provide links to resources, such as images, web pages, or other concept maps*

Novak & Canas (2008) developed a concept-map, a way of representing relations between ideas, images or words. The structure that a sentence diagram represents the grammar of a sentence, a road map represents the locations of highways and towns, or a circuit diagram represents the workings of an electrical appliance, for example, are presented as a connection of concepts with labeled arrows, in a downward-branching hierarchical structure. The relationship between concepts can be articulated in linking phrases, such as "gives rise to", "results in", "is required by" or "contributes to" (Figure 8.2). The concept maps are constructed to reflect the organization of the declarative memory system. They grow within a context frame defined by explicit focus questions, such as "what is ..?", "what caused ..?" or "what is required by ..?".

8.2 Visual Language-based Sketching Interface

Sketching is already known as a powerful means of interpersonal communication. People draw, point, mark, highlight, underscore and use other gestures to help disambiguate what they are saying. The sketching usually involves a combination of interactive drawing plus language interaction. The drawing carries the spatial aspects of what is to be communicated. The linguistic interaction provides a complementary conceptual channel that guides the interpretation of what is drawn. The verbal description that occurs during drawing compensates for inaccuracies in drawing. This because most people are not artists. Even artists cannot produce, in real time, drawings of complex objects and relationships that are recognizable solely visually. A computational model of sketching would help characterize the ambiguities in language and help us better understand how to overcome them. Here, a first step towards such a model is introduced.



There is a car accident.

A car and a truck are involved in a collision at 17:27 PM.
The car accident causes some big explosions.
The explosions generate fire and toxic smoke.

Figure 8.3: *The interface of the free visual language-based communication*

Instead of hand-drawing of objects and their relations, a free and natural way of creating a spatial arrangement of predefined graphics symbols (such as icons, lines, arrows and shapes) to represent concepts or ideas is proposed. Figure 8.3(a) shows the visual language-based interface, which has been developed for reporting observations in the field of crisis management. The interface provides a drawing area where users can attach and re-arrange icons, lines, arrows and ellipses on to describe the observations of crisis situations. An icon can be selected from three options: (a) the icon menu in which the icons are grouped based on their concepts, (b) the icon prediction results, which are calculated by adapting an n-gram word prediction technique (see Section 4.4), and (c) the search engine, which finds icons based on a keyword. The interface supports the users with mechanisms to draw and delete ellipses and links. As messages are constructed, the internal representation of the state of the world is continuously analyzed, created or adapted. At the same time, the scenario that matches the messages is also constructed.

8.3 A Grammar Free Visual Language

Using icons to represent concepts, the (grammar-) free visual language provides a way of representing the world phenomena using a spatial arrangement of icons. The icons can also represent structural units of the visual perception, such as direction, color, motion, texture, pattern, orientation, scale, angle, space and proportion. Similar to the concept- map, the relationship between icons can be formulated using lines (or

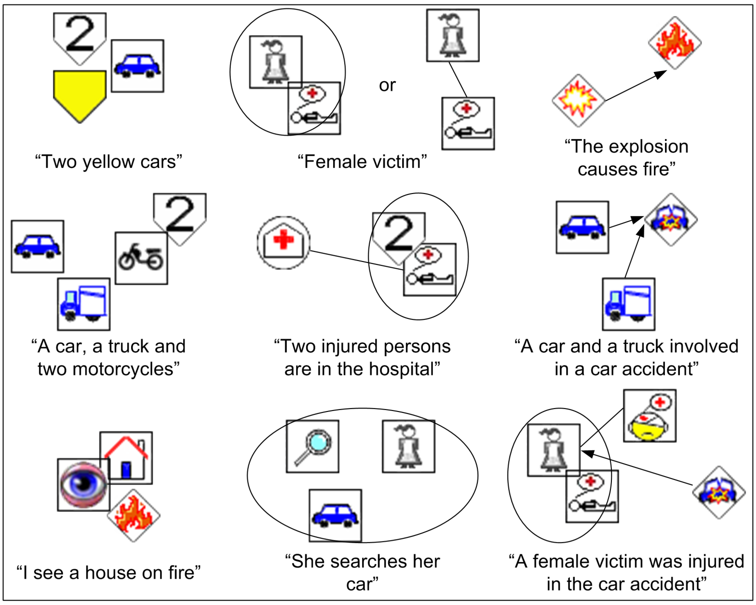

“Two yellow cars”
or
“Female victim”
“The explosion causes fire”
“A car, a truck and two motorcycles”
“Two injured persons are in the hospital”
“A car and a truck involved in a car accident”
“I see a house on fire”
“She searches her car”
“A female victim was injured in the car accident”

Figure 8.4: *Examples of the free visual language-based sentences; The order of the icons and their relations does not have to be linear*

arrows). However, the order of the icons is not necessary to be linear or hierarchical.

To create a message, icons can be connected using arrows and lines. The arrow can specify the direction of the relationship, such as causal, temporal and possession. While the lines can be used to specify undirected relationships between icons, such as correlation and conjugation. Grouping some icons that represents a close relationship can be expressed using ellipses around the related icons. Another way is simply by placing them close to each other and away from other "unrelated" icons. All of these free arrangements can create a meaningful visual language-based "sentence" that represents a context. Figure 8.4 shows examples of messages based on the third proposal of visual language.

8.4 The Law of Proximity

The current work explores a method to interpret the way in which people relate certain concepts to describe events. Here, we focus on the law of proximity from the gestalt psychologist. This law states that the brain more closely associates objects close to each other than it does when two objects are far apart. According to this theory, spatial or temporal proximity of elements may induce the mind to perceive a collective or totality. One of the learning points from this theory is that not only because of human's inclination of wanting to be neat and organized, by grouping objects together, we are also making room for other interesting information to be added to a layout that would not fit otherwise. Therefore, in knowing about this law, we can interpret visual messages being conveyed, because of the relationships underlying the grouping of elements within the messages.

An agent-based interpretation method has been modeled to interpret the spatial arrangement of visual symbols on the user's workspace. Taking icons, lines, arrows and ellipses as visual elements in focus, using the law of proximity, the model consists of cooperative and competitive interactions of the visual structure building agents driven by plausible spatial grouping pressures. Incoming visual symbol inputs are treated as possible relations to be interpreted in conjunction with the emergence of higher-level structures. This relations-building mechanism is carried through for obtaining context sensitivity and interpretative capabilities.

Let a user workspace W contains a list of visual symbols, such as icons I, lines L, arrows A and ellipses E. An icon $w \in I \cup W$ represents a concept object α in the world model Ω. It is denoted as $\alpha = \{t, [x_1 : p_1 = f_1, \ldots, x_n : p_n = f_n\}$ where $t \in [p_1 \ldots p_n] \in T$ are the types, $[x_1 \ldots x_n] \in P$ the properties of α and $[f_1 \ldots f_n] \in F \cup \Omega$ the values of the properties. For $[w_1 \ldots w_n] \in I \cup W$, let an arrow is denoted as $a(w_1, w_2)$, which means w_1 is connected by $a \in A \cup W$ to w_2; a line is denoted as $l(w_1, w_2)$, which means $w_1, w_2 \in W$ are connected by $l \in L \cup W$; and an ellipse is denoted as $e(w_1, w_2, ..., w_m)$, which means $[w_1 \ldots w_m]$ are intersected with $e \in E \cup W$.

For two visual symbols $v_1, v_2 \in W$ consists a gravity function δ, which returns

$Boolean$. This function is defined as $\delta(v_1, v_2) = true$ if v_1 is placed next to v_2 and there is not any other symbol in between, otherwise it returns $false$. The relation of concepts $\alpha, \beta \in \Omega$ is defined as:

$$\alpha \Rightarrow \beta = \left\{t_\beta, [y_j : q_j = g_j | q_j = t_\alpha, g_j = \alpha\right\} \tag{8.1}$$

where $\alpha = \{t_\alpha, [x_i : p_i = f_i, \ldots, x_n : p_n = f_n]\}$, $\beta = \left\{t_\beta, [y_j : q_j = g_j, \ldots, y_m : q_m = g_m]\right\}$, $\alpha \neq \beta$ and $(\alpha \Rightarrow \beta) \neq (\beta \Rightarrow \alpha)$. This function yields a cue of a plausible visual structure. If $w_\alpha, w_\beta \in I \cup W$ represents $\alpha, \beta \in \Omega$, the function is applied in the following cases (see their illustrations in Figure 8.5):

1. ***If connected by an arrow***:

$$\alpha \Rightarrow \beta = a(w_\alpha, w_\beta) \cap q_j = t_\alpha \tag{8.2}$$

where $[q_1 \ldots q_m] \in T_\beta \in T$ and $t_\alpha \in T$.

2. ***If connected by a line***:

$$\alpha \Rightarrow \beta = l(w_\alpha, w_\beta) \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.3}$$

where $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

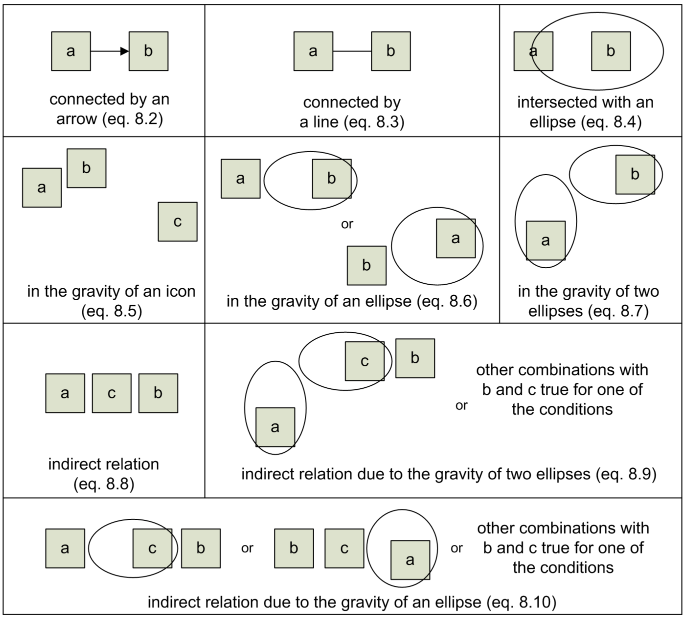

a
b
connected by an arrow (eq. 8.2)
a
b
connected by a line (eq. 8.3)
a
b
intersected with an ellipse (eq. 8.4)
b
a
c
in the gravity of an icon (eq. 8.5)
a
b
or
a
b
in the gravity of an ellipse (eq. 8.6)
b
a
in the gravity of two ellipses (eq. 8.7)
a
c
b
indirect relation (eq. 8.8)
c
b
a
or
other combinations with b and c true for one of the conditions
indirect relation due to the gravity of two ellipses (eq. 8.9)
a
c
b
or
b
c
a
or
other combinations with b and c true for one of the conditions
indirect relation due to the gravity of an ellipse (eq. 8.10)

Figure 8.5: *Illustrations of interpreted relations between $icon_a$ and $icon_b$*

3. ***If Intersected with an ellipse***:

$$\alpha \Rightarrow \beta = e(w_\alpha, w_\beta) \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.4}$$

where $e \in E \cup W$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

4. ***If close to each other***:

$$\alpha \Rightarrow \beta = \delta(w_\alpha, w_\beta) \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.5}$$

where $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

5. ***If in the gravity of an ellipse***:

$$\alpha \Rightarrow \beta = (\delta(w_\alpha, e(w_\beta)) \cup \delta(e(w_\alpha), w_\beta)) \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.6}$$

where $e \in E$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

6. ***If in the gravity of two ellipses***:

$$\alpha \Rightarrow \beta = \delta(e_1(w_\alpha), e_2(w_\beta)) \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.7}$$

where $e_1, e_2 \in E$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

7. ***If indirect relation***:

$$\alpha \Rightarrow \beta = \delta(w_\alpha, w_\gamma) \cap \gamma \Rightarrow \beta \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.8}$$

where $w_\gamma \in I$, $\gamma \in \Omega$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

8. ***If indirect relation due to the gravity of two ellipses***:

$$\alpha \Rightarrow \beta = \delta(e_1(w_\alpha), e_2(w_\gamma)) \cap \beta \Rightarrow \gamma \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.9}$$

where $e_1, e_2 \in E$, $w_\gamma \in I$, $\gamma \in \Omega$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

9. ***If indirect relation due to the gravity of an ellipse***:

$$\alpha \Rightarrow \beta = (\delta(w_\alpha, e(w_\gamma)) \cup \delta(e(w_\alpha), w_\gamma)) \cap \beta \Rightarrow \gamma \cap (q_j = t_\alpha \cup p_i = t_\beta) \tag{8.10}$$

where $e \in E$, $w_\gamma \in I$, $\gamma \in \Omega$, $[p_1 \ldots p_n] \in T_\alpha \in T$, $[q_1 \ldots q_m] \in T_\beta \in T$, and $t_\alpha, t_\beta \in T$.

When the structures of the visual symbols are recognized, the interpretation of the messages can emerge as adaptations to the outcomes of interactions with properties of reported concepts (defined by the ontology). These interactions are then inferred by known crisis scenarios (defined by scripts) for forming the definition of such interactions (see Chapter 12). The interpretation results a coherent and context dependent world model as the manifest interpretation of the arrangement and the interaction of visual symbols in a user workspace. In the domain of reporting observations of a crisis situation, the world model can be used to form (natural language) crisis scenarios as feedback to the user on his/her visual symbol arrangements. A scenario is generated by simple concept-name recognition and substitution of the property-names using their values controlled by an extended XML format (see Section 13.3).

8.5 System Design

Figure 8.6 shows the class diagram of the free visual language-based interface. The FFree class handles the interaction with users, which includes handling the message creation, the display of the scenario and the display of the icon prediction results. This class has a ReportProcessor class that handles all active Report objects. Each Report object contains a Free2D object that stores the spatial arrangement of visual symbols used for constructing the report. The ReportProcessor class uses the fusion module (see Chapter 12) to interpret the input. The module receives a spatial arrangement of visual symbols and results a list of linked concepts. This list represents the world model of the current situation as the manifest interpretation of the communicated message. In this process, the fusion module uses the ontology module and the script module (see Chapter 9). The list of linked concepts is sent to the fission module to construct the natural language representation of the current crisis scenario (see Section 13.3). To create the scenario, this class uses the aiml module from the library that consists templates for developing a natural language crisis scenario in an extended XML format.

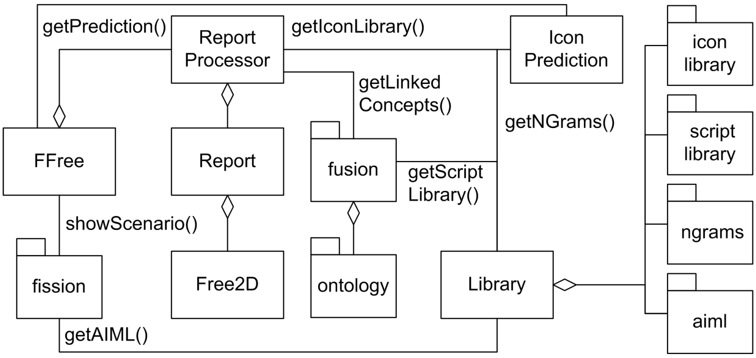


Figure 8.6: *The class diagram of the free visual language-based interface*

The free visual language-based interface is served as the interface of the research demonstrator reported in this thesis. The system design of the interface becomes one of the module of the demonstrator. Therefore, the user testing of this interface is included in the user testing of this demonstrator, which is discussed in Chapter 14.

Chapter 9

Knowledge Representation

In which the proposed knowledge structures of the adaptive HHI framework, such as the ontology and scripts, are described.

Knowledge is the appropriate collection of information that is presented within a particular context (Ackoff, 1989). In the HCI domain, knowledge serves as information storage that is relevant to the interaction between a user and an application. To "understand" the meaning conveyed in the interaction, knowledge must be modeled as the structured record of the user's world. To represent a logical and abstract view about the world, in this research, ontology-based knowledge representation is selected. Knowledge representation presented in this chapter is defined for a specific problem domain, that is *the field of crisis management.* To define the knowledge representation of other domains, the defined research methodology and design process can be followed since the ontology enables us to use/reuse other domain knowledge.

9.1 The Conception of the World

In most general sense, communication involves exchanges of representations. These representations are known as concepts. Concepts are humans' internal mental models and representation of themselves, the outside world and of things with which they are interacting (Perlovsky, 1999). These concepts help to predict and understand the communicated messages. The results of this conceptual understanding are actions, to the outside world or inside their mind.

Knowledge in a computing domain is a structured record of relevant entities, processes and events that have some effects on user-system interaction. The development of knowledge representation is built up based on both initialization and interaction with the user and environment. The modeling of knowledge representation constitutes a major challenge, in such a way, the information structures should share a common concept space and are able to fulfill the context-aware requirements.

Ontology consists of a set of concepts, a set of relationships and a set of axioms that describe a domain of interest; complemented with two partial orders defining the concept hierarchy and relation hierarchy respectively. With an ontology the concepts that will be used, can be precisely defined. The ontology provides mechanisms not only for identifying the relationship between concepts but also for identifying the context information about a concept. Classes in the ontology are specified according to how they stand in relation to other classes. This allows us to place the appropriate rules on their respective concepts. The common concepts should be identified for example, the relation "x caused y". Then, logical relationships to formulate rules about those common concepts can be used. For example, "if x caused y, then x temporally precedes y". In this way, ontology representation can give user-system interaction enough expressiveness and facilitate the reasoning.

9.2 Corpus and Expert-based Knowledge Engineering

Consider that during crisis situations crisis workers in the field have to communicate with people in the crisis center in order to achieve a most effective crisis response. Figure 9.1 shows a schematic view of a crisis situation. A crisis responder in the field (agent) is an autonomous participant that can report observations. Such sensed data, what observers see, hear, smell and even feel or experience, is transformed into reports for others. At any time, agents are associated with a location in the geographical space. The crisis center collects the reports and forms a global view about the crisis. An intelligent system can help them to calculate the most probable causes, predict the impact of the crisis and give advices based on the world model. Further, knowledge of the world is used for supporting the decision making process and executing actions.

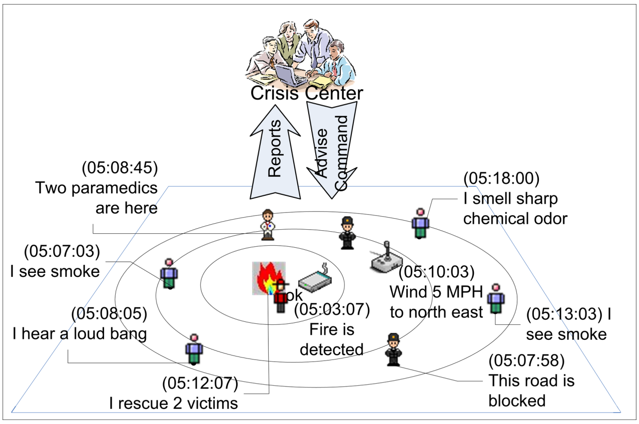

Crisis Center
Reports
Advise
Command
(05:08:45) Two paramedics are here
(05:18:00) I smell sharp chemical odor
(05:07:03) I see smoke
(05:10:03) Wind 5 MPH to north east
ok
(05:03:07) Fire is detected
(05:08:05) I hear a loud bang
(05:13:03) I see smoke
(05:07:58) This road is blocked
(05:12:07) I rescue 2 victims

Figure 9.1: *Disaster dispersion occurs in the world; agents in the field report their observation*

Following the formulated scenario, the world knowledge formation based on user reports is investigated. To understand how people report crisis situations, three different data sources have been analyzed: (1) over 100 dialogues from 911 recording transcriptions (SFMuseum, 1989; WNBC, 2002), (2) news articles with six different crisis topics (such as fire, bombing, flooding, tsunami, earthquake, and terrorist attack), and (3) interview results with experts from the Southern Rotterdam Fire Department and the crisis center of the Rijnmond environmental agency (DCMR) (Benjamins, 2006). In addition to these, research in developing visual language-based interfaces (Fitrianie, 2004; Fitrianie et al., 2006, 2008b) has formulated knowledge methods for creating visual language-based messages, methods for interpreting the message and the design of visual language-based interfaces. The findings provide requirements of necessary concepts for developing a knowledge representation. The overall studies serve a number of purposes: (a) developing a list of possible crisis scenarios, (b) developing a list of concepts that play an important part in crisis situations, (c) analyzing relations between concepts, (d) designing and developing corresponding icons and (e) designing an interface for reporting observations.

The analyses results show that a hypothesis concerning a crisis situation is generated from the information extracted from the reports arriving from different observers during the crisis. The hypothesis is based on considerations of a limited set of crisis scenarios, such as fire, explosion, chemical contamination and the like. Each scenario has associated features, some of which can be important factors to distinguish one scenario from another. Each feature mainly is described as the result of human perception, such as sound, vision, odor, and feeling. A report usually focuses on certain keywords and specific terms.

With the information from the recorded corpora and experts, a set of predefined crisis scenario and their associated features (concepts) were formulated. Next, the formulated scenarios were tested to make sure that the concepts associated with each scenario were discriminative enough so that each of these scenarios was identifiable by a unique set of concepts. Finally, the concepts were further developed into an ontology, while a selection of the scenarios were developed into scripts. A script represents the chain of events to identify a possible scenario. In the final step, corresponding icons of selected concepts were developed. Since visual symbols may have different or even no meaning outside of their context, the ontology is employed to represent context that binds verbal and icons together. To avoid ambiguous meanings created by the relation between the icons and words, a verbal context that can link both visual and verbal thoughts is designed together to form an icon that can be remembered and recalled. User tests were conducted to ensure that the "universal" meaning of the knowledge representation can be conveyed as intended by the designer of the icons.

9.3 Ontology-based Knowledge Structure

The ontology is defined as a body of knowledge of user model, task model and world model. Figure 9.2 shows the model of the three knowledge resources. The world model is a symbolic representation of the real world environment hierarchically structured as a directed graph. The user model is knowledge about the users. It includes information about their profile, their emotional state and their communication device. The task model contains general ideas of how tasks should be done. The knowledge form the awareness of the state of the user, the interaction and the current situation.

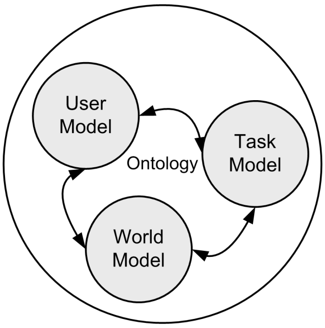


Figure 9.2: *The schematic view of the knowledge structure*

9.3.1 A Graph Representation of the World

Crisis management, for events such as natural disasters, technology failures, aviation accidents and acts of terrorism, relies upon teams of people who must collaboratively derive knowledge from geospatial information. A crisis event is viewed as a chain of temporal specific events and a group of dynamic objects (including actors with a specific role) in actions at a certain location in the world. This view represents a dynamic context of the world model. The dynamic objects can be defined as an individual or a group of entities at a certain location. This type of entities is viewed as a dynamic object or event or relation if it has dynamic locations for a certain range of time. It also may appear at a certain moment and disappear from the view at the next moment. For example, fire, fireman and firetruck. While the static context of the world model is actually the geographical information about the crisis location. Both dynamic and static contexts are organized into two layers (Figure 9.3). Since both data sets are georeferenced, the entities of both contexts have real-world locations and overlay one another. In addition, both knowledge may have direct links conceptually to icons on the user interface (Figure 9.4(a)).

The information of the static contexts of the world is retrieved from a geographic database (geo-database). An object in the geo-database represents a feature of a real-world entity, for example a parcel, a building, a street, a river or a streetlight. It includes the information about features of a building, such as living rooms, kitchens, stairs, doors and the like. The feature geometry can be described using relationships between nodes (geographical points) and edges (connections between two nodes) to develop spatial relationships, such as topologies and networks. Topology is used to manage common boundaries between features. The networks describe a connected graph of geo-objects that can be traversed. This is important for modeling pathways and routing navigations. For example, a street's features represent edges that connect

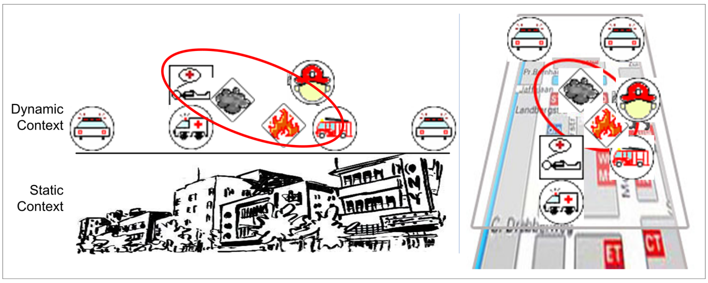


Figure 9.3: *A schematic view of static and dynamic contexts of the world overlay one another; side view (left) and top view (right)*

at their endpoints. The geo-database also includes geographic information, such as addresses, place names, geo-processing information (such as area measurement) and cartographic information.

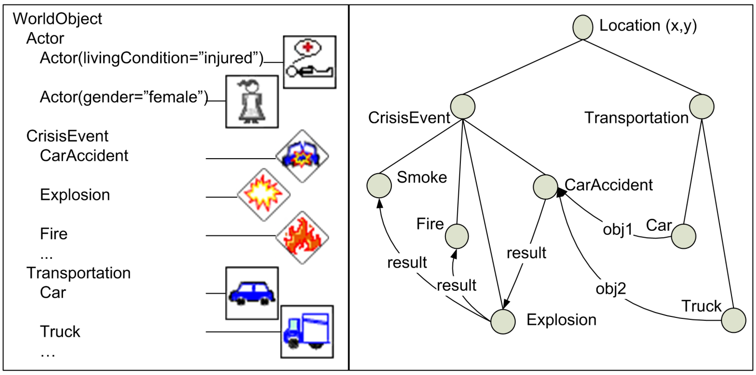


Figure 9.4: *(a) The WorldObject class Taxonomy: an icon is an instance of a class in the ontology and (b) a graph-based representation of "on (x,y), an accident between a car and a truck results in an explosion; the explosion causes fire and smoke"*

The world model contains fundamental information about a crisis situation in a certain perspective-area. The geospatial knowledge of the crisis event is represented with topographical data using graphs for data modeling. The graph connects its nodes based on their approximated spatial coordinates in the world. The nodes not only contain specific information of the individual, but also their current status (such as living condition, hazardous level, dynamic spatial state and so on), their temporal information (such as frequency and time point) and their spatial information (such as current location, origin, destination and path). The arcs represent the hierarchy of groups of individuals. Another arrow shows a relation between individuals (such as `containOf`, `result`, `causal` and so on). Figure 9.4(b) shows an example representation of a traffic collision event.

9.3.2 User Model

The user model is defined as a dynamic model of users that are registered in the user database. The information of each user is stored based on their identity, interaction time, current communication media and modality(ies), and current location in the world. A user in this user model can be a part of actors in the world model, which is referred by its user id. Most of the information will be used by the fission module (Section 13.3) to define appropriate outputs.

There are two types of users, such as civilians and professionals. Professional users are classified to fire-fighters, polices, paramedics, military, and authorities. Other users are civilian. A professional user has a rank and role. This information will be used to filter the right information contents for the right user. Some information in the ontology may include a predefined flags to specify which information is available for which user's role. Knowledge about previous user system interaction will be used as a reference to select the most up to date information to each individual user. The way of this information is presented to a user depends on the information about the user's current emotional state. The content of the information presented also depends on the location of the user. This context information will be used to define the perspective area of the world knowledge of this user, since the perspective view of a user may differ from the perspective area of the world model in general. It can be computed using a set of policies based on the models of natural physical phenomena and geographical area of crisis events.

At least one communication channel is used by a user. A communication channel has one or more communication infrastructures, which contain information about communication media and modalities. This information will be used for selecting the output modality to convey the output messages. A set of negation policies is utilized based on the characteristics of modality services of the user's communication device. The selection of the right policy also depends on the type and the availability of the modalities.

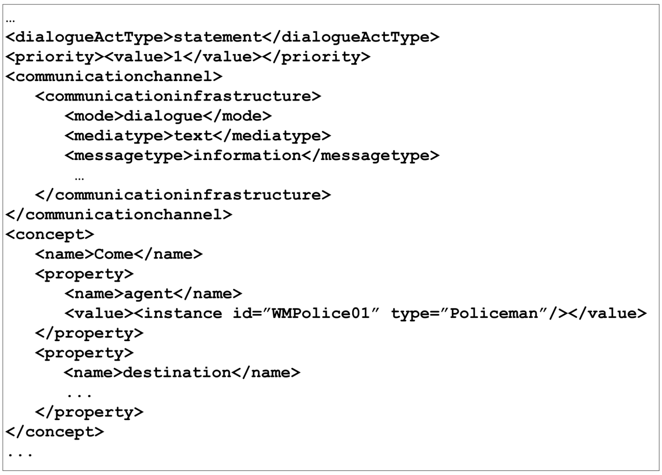
9.3.3 Task Model

In the proposed framework, the user system interaction flow is controlled by an IM module (see Section 13.2). This module selects a set of appropriate communication acts and sends them to a fission module (see Section 13.3). Table 9.1 shows ten possible communication acts that have been implemented.

Figure 9.5 shows an example of a communication act message. A communication act contains a priority level, a set of output communication channels, timing, a link to other processes and reference to the world model. It is possible that in a multi-user system, the IM module also defines which user(s) will receive such information. The fission module uses the selected communication act as one of the criteria for constructing the output.

Table 9.1: *The implemented communication acts*

Communication Act	Description
`statement`	To notify about an event
`request`	To ask a specific action
`ask`	To request a specific information
`clarify`	To ask a clarification
`acknowledge`	To display incoming information
`highlight`	To highlight (on the display or repeat) a certain information
`removal`	To remove specific information from displays
`affirmation`	To notify its agreement to the user's action
`negation`	To notify its disagreement to the user's action
`scenario`	To display current crisis scenario



```
…
<dialogueActType>statement</dialogueActType>
<priority><value>1</value></priority>
<communicationchannel>
   <communicationinfrastructure>
      <mode>dialogue</mode>
      <mediatype>text</mediatype>
      <messagetype>information</messagetype>
       …
   </communicationinfrastructure>
</communicationchannel>
<concept>
   <name>Come</name>
   <property>
      <name>agent</name>
      <value><instance id="WMPolice01" type="Policeman"/></value>
   </property>
   <property>
      <name>destination</name>
      ...
   </property>
</concept>
...
```

Figure 9.5: *A communication act: "notifying user that a policeman is on their way to a destination"; the policeman is referred to by an id in the world model*

9.3.4 System Design

The Crisis Management Ontology was designed to capture the graph structure of the world model, which is stored in the `ontology` module. The upper level ontology of the SmartKom, M3L (Figure 9.6) (Gurevych et al., 2003), is adopted for the upper layer of the Crisis Management Ontology. The current implementation contains only classes that are necessary for the user system interaction and the message interpretation. It consists of 127 classes connected to 110 icons (see Appendix A.2).

The `PhysicalObject` class is divided into classes that are necessary to describe the world model and the user model, such as:

1. The `AreaPerspective` class refers to a logical location based on the world co-

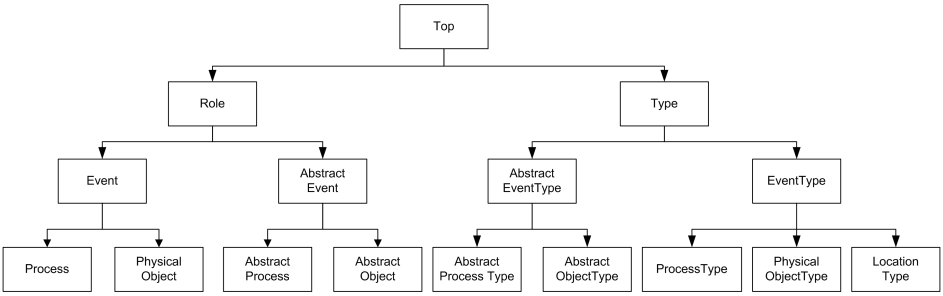


Figure 9.6: *Upper ontology of the crisis management ontology*

ordinate where a crisis event occurs, an object placed or an action executed. This class represents the top level of the geospatial graph.

2. The `WorldObject` class handles entities that are involved in a crisis event. It consists of the `DynamicObject` class and the `StaticObject` class.

3. The `User` class handles information about registered users.

4. The `Communication` class handles information about the channel and the infrastructure of the communication device of a user. Information about the relationship between a user and one or more communication infrastructures is stored in the `CommunicationChannel` class and the end point information is handled in the `CommunicationInfrastructure` class.

The `WorldObject` class and the `User` class are described in the following. The next is, in addition, the `Process` class and the `Type` class are also presented.

The WorldObject Class Taxonomy

Figure 9.7 shows the class taxonomy of the `WorldObject` class. The dynamic content of the world is defined by the `DynamicObject` class. The subclass of this class includes real entities, such as: (a) the `Actor` class, which refers to a human with a certain role that may be a user, (b) the `CrisisEvent` class, which refers to the type of crisis events, (c) the `Substrate` class, which refers to a chemical and non-chemical substance, (d) the `Transportation` class, which refers to the type of vehicles, and (e) the `Tool` class, which refers to an instrument to work with.

Binary relations between concepts are defines as "properties". For example, the `WorldObject` class contains `location`, `beginTime` and `endTime`. A property can be an instant of another class. For example, the `Substrate` class is described by the `Size` class, the `Color` class, and the `Weight` class. Other properties contain information about the state of objects. For example, the `Actor` class has `isTrapped`, `livingCondition`, `rescuedLevel` and `dynamicSpatialState`. Such relations

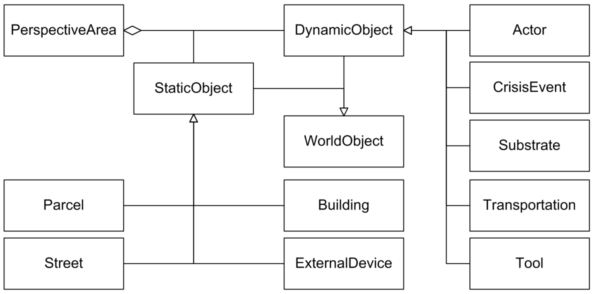


Figure 9.7: *The taxonomy of the WorldObject class*

make the properties purely relational. Other properties may be functional or transitive. Examples of functional slots are `age`, which is required by a `Person` object, and `location`, which is required by a `DynamicObject` object. An example of a transitive property is `causal`, which has the inverse `result`.

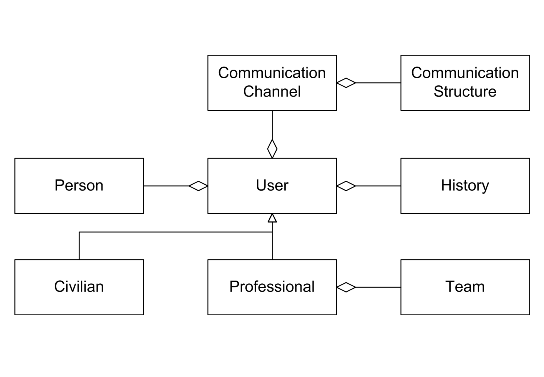
```
<house name=”House A”>
  <capacity>6</capacity>
  <topology>
     <node name=”nodeA”><x>40</x><y>34</y></node>
     <node name=”nodeB”><x>112</x><y>28</y></node>
     <node name=”nodeC”><x>40</x><y>134</y></node>
     <node name=”nodeD”><x>120</x><y>34</y></node>
     <edge name=”edge1"><node1>nodeA</node1><node2>nodeB</node2></edge>
     <edge name=”edge2"><node1>nodeB</node1><node2>nodeC</node2></edge>
     <edge name=”edge3"><node1>nodeC</node1><node2>nodeD</node2></edge>
     <edge name=”edge4"><node1>nodeD</node1><node2>nodeA</node2></edge>
  </topology>
  <address>
     <street>Prins Alexanderstraat</street>
     <number>97</number>
     ...
  </address>
</house>
```

Figure 9.8: *A record of a house*

The `StaticObject` class subsumes the static entities, such as `Street`, `Parcel`, `Building` and `ExternalDevice`. Examples of the `ExternalDevice` class are a sign, valve, door, camera, detector or actuator. Figure 9.8 is an example of a record structure in the geo-database about a house. The `Topology` class is used to manage common boundaries between features. It is described using relationships between nodes and edges. A node is a (x, y) position in the world and an edge is a link between two nodes.

The Class Diagram of User

Figure 9.9 shows the class diagram of the user model. The `User` class is divided into the `Civilian` class and the `Professional` class. It has a dynamic location, active communication channels, current status and conditions, and an emotional state. The `Person` class is defined as an independent entity of a user that stores most static information about the user. The dynamic information about users is stored in the User class. Every interaction that relates with a user is archived in the History class. The User class has at least one CommunicationChannel object. This object contains one or more CommunicationInfrastructure objects.



```
Process
   GeneralProcess
       TransferAct
           Give
       ...
   MentalProcess
       SensorAct
           See
       ...
   PhysicalProcess
       RescueAct
           Search
           Evacuate
       ...
   SocialProcess
       CommunicationAct
       ...
```

Figure 9.9: *The class diagram of User (left) and the taxonomy of the Process class (right)*

The Process Class Taxonomy

The Process class subsumes the same extensive inventory of processes as the ontology of SmartKom (Gurevych et al., 2003), such as: (a) the GeneralProcess class, (b) the MentalProcess class, (c) the PhysicalProcess class, and (d) the SocialProcess class. Figure 9.9 shows parts of the taxonomy of this class. The CommunicationAct class is a subclass of the SocialProcess class, which contains classes for communication acts. The type of rescue and responding activities are defined as the subclasses of the RescueAct class. For example, the Search class, the PutFire class, the Evacuate class and the like. The PhysicalProcess class is the super class of these classes. The binary relations between classes of the Process class were designed using thematic roles, such as agent, patient, instrument, source, destination and so on. For example, the Search class has agent and patient as the instances of the Actor class and the location as the instance of the Location class.

Abstract Classes and Type Classes

In the AbstractEvent class, the AbstractObject class subsumes the Set class, the AbstractNumber class and the AbstractRepresentationalObject class. The AbstractRepresentationalObject class describes relations, directions and attributes of objects (Table 9.2). The Attribute class is used to define objects and events. The Direction class and the SpatialRelation class are largely defined based on English preposition. The AbstractObjectType class subsumes the non-role counterpart classes of the AbstractObject class (Table 9.3). The structure for representing data is defined under the `DataStoringType` class, while the information and constraints of media to represent a message is stored in the `ModeType` class. The `PhysicalObjectType` class subsumes the non-role counterpart classes of the `PhysicalObject` class. In the current implementation, this class subsumes the `Team` class, which refers to a role that a user may perform.

Table 9.2: *The taxonomy of the AbstractRepresentationalObject class*

AbstractRepresentationalObject	Attribute
Angle	Address
Attribute	Color
Direction	Emotion
CompassDirection	Gender
MovementDirection	Language
Measurement	Location
Size	Priority
Frequency	Property
Weight	Rank
Preference	State
Relation	CapacityState
Scalar	DynamicSpatialState
SpatialRelation	HazardousLevel
TemporalRelation	IsTrapped
Causality	LivingCondition
...	RescuedLevel
	SubstrateCondition

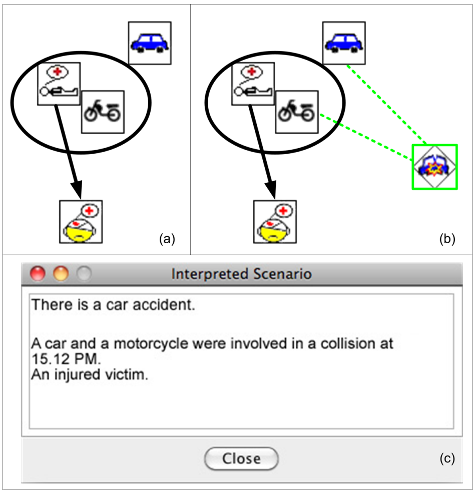
The `LocationType` class replaces the upper level - `Location` class of M3L to distinguish it with the `location` property of the `WorldObject` class. In particular, the `SpacePoint` class in this taxonomy refers to the coordinates of an object in a space.

Table 9.3: *The taxonomy of the subclasses of the Type class*

AbstractObjectType	LocationType
DataSotringType	SpatialEntity
AttributeValue	Path
Latice	Topology
Literal	SpaceInterval
ModeType	Height
Text	Length
VisualLanguage	Width
Audio	SpacePoint
Number	TemporalEntity
PhysicalObjectType	TimeInterval
Team	DateTime

9.4 Dynamic Scripting

In film productions, a storyboard has been used to provide a visual layout of events as can be seen through the camera lens. It is the layout and sequence in which the user or viewer sees the content or information. One advantage of using storyboards is that it allows the user to experiment with the storyline using different ideas of events and interest. In experiments, we have conducted such an storyboard approach triggers parts of the movie scripts in the minds of the movie makers to develop possible crisis scenarios as the starting point of developing the knowledge representation. Together, the formulated knowledge on how people report crisis situations (from different data sources) and the storyboard-like approach have been served to develop a dynamic scripting approach that is used to interpret a report dynamically during the message creation. Instead of trying to fit a user's interpretation of observations into sensible reports, this approach uses possible crisis scenario as the starting points of the reports.



There is a car accident.

A car and a motorcycle were involved in a collision at 15.12 PM.
An injured victim.

Figure 9.10: *An example of (a) a user input, (b) a proposed new concept (green border) and links (dashed green lines) based on the user input, and (c) the resulted interpretation*

While composing an observation report, a user of an observation report interface may start from identifying concepts that the user perceived from the environment. By employing a form of case-based reasoning modeled after the human's method of recognizing situations, this interface can help this user by: (a) proposing new concepts and their composition in the user's message adaptively and (b) creating possible scenario from the compositions of the input concepts so far. As the possibilities for concepts to be presented in the report can be very large, the interface can help to make selection of concepts to be included in the message. This selection of concepts is determined by a rule-based selection algorithm powered by competing scripts, hence the term dynamic scripting (Yang & Rothkrantz, 2007). New concepts may become active due to the interaction of the concepts in the user's input. Figure 9.10 shows the reasoning results using the dynamic scripting approach on a free visual language-based interface. In this sense, the interface guides the user to compose his/her observation messages into a possible scenario.

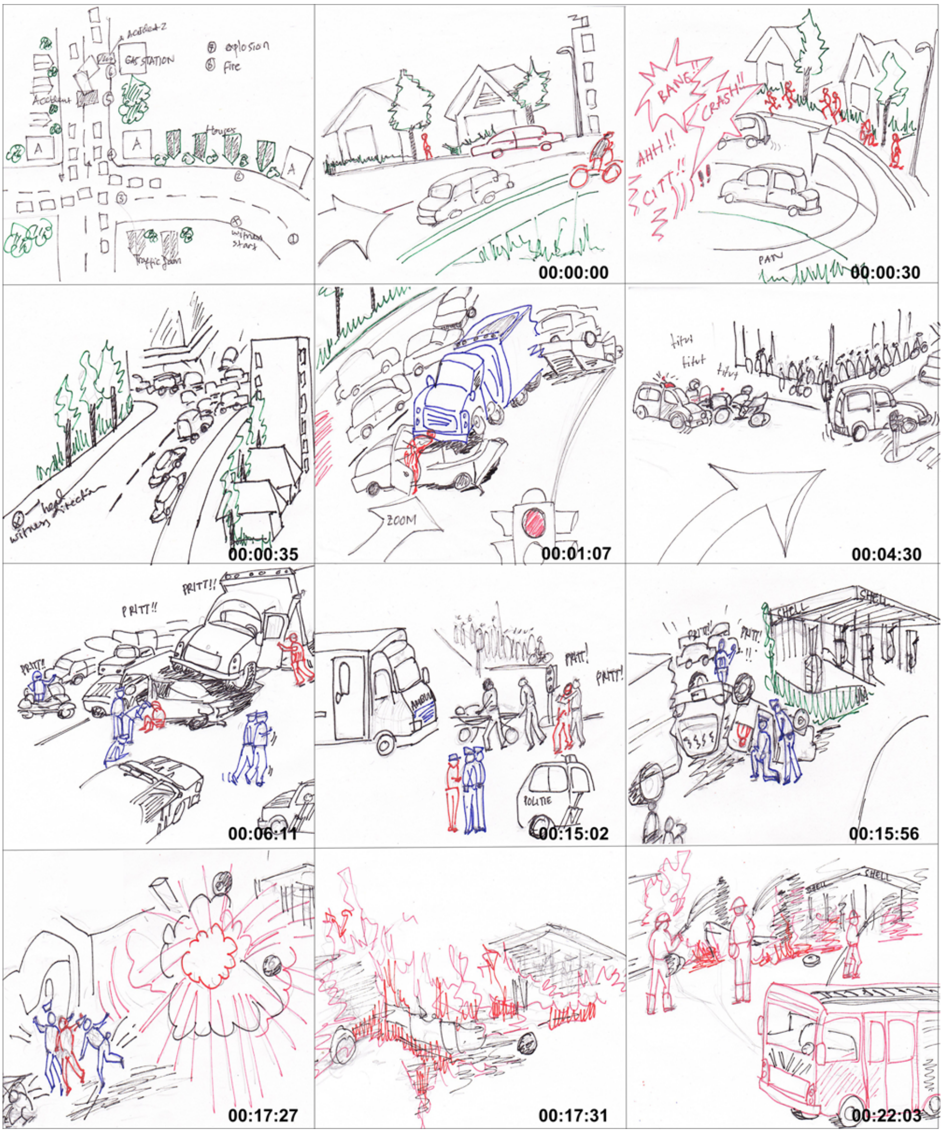


Figure 9.11: *An example of a storyboard: "witnessing a car accident"*

9.4.1 Case-based Scripts

According to Schank (1973), the sequence of events that constitute knowledge about situations are stored in human memory as scripts. The scripts are triggered by the observation of special features or events occurring in the environment. In the beginning when a small number of features are observed, many competing scripts may be in the mind. As the additional information from further observation and the reasoning of possible situations have been discovered, we can arrive at a final hypothesis.

Figure 9.11 shows the illustration of a traffic accident using a storyboard-like drawing. At initial events, we hear screams and loud noises. Then, we notice that some vehicles are collided, some people got injured and traffic jam occurs. In such observation, we may already have concluded what has happened. Shortly after, we can expect the ambulance and police to arrive. When we hear explosions and sense fire, we know that the situation become dangerous also for the witnesses. We also know that firefighters are on their way.

```
<script name="CarAccident" type="acknowledge">
  <case name="victim" logic="and" cost="0.9">
    <condition operator="biggerEqualThan" function="getTotalFrequency()">
       <concept name="Transportation" propertyName="frequency"
                propertyType="Frequency.valueInt:int"/>
       <constant value="1" type="int"/>
    </condition>
    <condition operator="equal">
       <concept name="Actor" propertyName="livingCondition"
                propertyType="LivingCondition.livingType:String"/>
       <constant value="injured" type="String"/>
    </condition>
  </case>
  <case name="professional" logic="and" cost="0.2">
    <condition operator="lessEqualThan">
       <condition function="getDistant()" unit="meter">
          <concept name="Actor" propertyName="location" propertyType="Location"/>
          <concept name="Ambulance" propertyName="location"
                   propertyType="Location"/>
       </condition>
       <constant value="1" type="double"/>
    </condition>
    <condition operator="biggerEqualThan" function="getTotalFrequency()">
       <concept name="Policeman" propertyName="frequency"
                propertyType="Frequency.valueInt:int"/>
       <constant value="1" type="int"/>
    </condition>
    ...
  </case>
  ...
</script>
```

Figure 9.12: *An Example of Scripts (in XML)*

In observing the situation, we can recognize or even identify a scenario with an event sequence consisting of many events from just the first few of these events. It seems that humans are very capable of recognizing and participating in situations that they have been experienced or observed before. Triggered by the key events being perceived by the human senses, such as sight, hearing and smell, a script is recognized without having perceived all the events that constitute the script. Humans have taken advantage of this property of such knowledge all the time, giving them the ability to rapidly interpret, recognize and participate in situations that resemble scenarios stored in their memory. This allows them to perform intelligent actions efficiently about frequently experienced events. Recognition errors can easily be corrected if more features become available.

A script is defined by some cases of events, which are distinguished by some conditions. Figure 9.12 shows an example of a script in XML format. In this example, a car accident event is acknowledged by some of cases of conditions. One of the cases depicts a situation when there is an injured actor in the area. A weight value (`cost`) is assigned on each case to show how important the case is. Usually, a case becomes important because it contains key concepts that make up the case's salience.

9.4.2 Rete-based Inference Method

Rete algorithm is a pattern matching algorithm for implementing production rule system (Forgy, 1990). Based on this approach, the scripts are built into a network of nodes. Every node (except the root) corresponds to condition part of a script (Figure 9.13). The path from the root node to a leaf defines a complete script's case. A node has a memory of concepts that satisfy the conditions of a case.

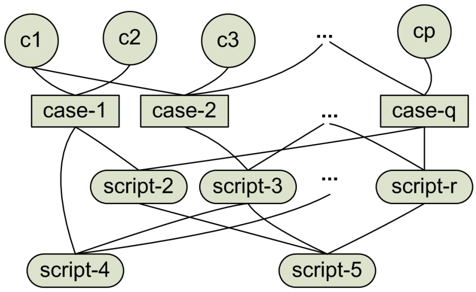


Figure 9.13: *An illustration of the rete-based network of scripts*

Only scripts that contain concepts entered specifically by the user, have a chance of being selected. As new concepts are asserted to other concepts, they propagate the network, causing nodes to be annotated when the known concept matches the conditions in a case of a script. When all conditions in a case for a given script to be satisfied, a leaf node is reached and corresponding script is triggered.

A conflict resolution strategy is applied to determine the order in which the scripts have to be selected. An *importance value* of a script is defined to measure how attractive the script is to be selected, which shows how many (salience) concepts are covered. It is calculated as the total of weight of all cases in a script. When a notion of the events occurring while interpreting the input message, certain scripts become more plausible then others. As more concepts are perceived, the evidence in favor of the plausible scenario increases. This in turn increases the plausibility of the corresponding of scripts. On the other hand, evidence might contradict a scenario. In this case another script becomes more plausible. Based on the ontology, certain concepts can be inferred from the existence of other concepts. For example, if there is a collision there is an accident. Using this ontology, derived concepts can thus be inferred even though not specifically entered by the user. As concepts in ontology have direct links to concepts in scripts, derived related concepts can increase the importance value of the related scripts, making them more plausible than other scripts. If there are more than one script, a script with the highest importance value will be selected, unless all values are below the low threshold. In such a case, none of scripts is selected.

9.4.3 System Design

Figure 9.14 shows the class diagram of the `script` module and the `Inference` class. The `script` module is aggregated into the `Library` class. The `ScriptLibrary` class receives a directory that contains XML files. It reads the files and extracts their contents into a list of `Script` objects, which contains at least one `Frame` object. The `Frame` class has one or more `Condition` objects, which are defined as an operation of two operands. Each operand is defined as a `Variable` object.

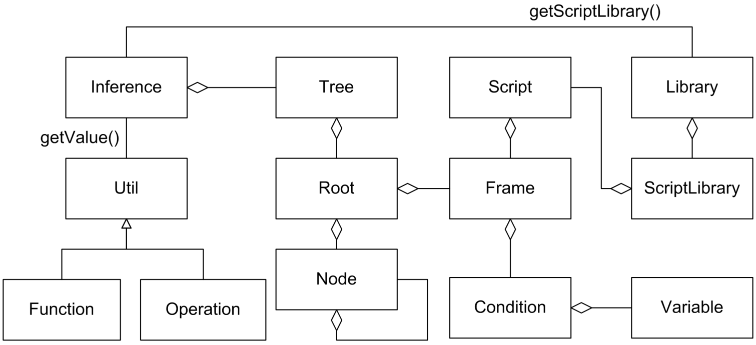


Figure 9.14: *The class diagram of the script module and the Inference class*

The `Inference` class is actually a part of the `fusion` module (see Chapter 12). It uses the scripts to build a rete-based `Tree` object. The `Tree` class contains a list of `Root` objects. A `Root` class is linked by a `Node` object to other `Root` objects or to other `Node` objects. These links create a network of scripts. An `Inference` object receives a set of interconnected concepts as its input. The concepts are treated as a list of facts and matched against the defined script using the `Function` class and the `Operation` class.

Chapter 10

Icon Database

In which the guidelines of icon design and presentation are discussed. The developed icon database in the domain of crisis management is presented.

A visual language-based interface naturally involves a large number of icons. In contrast to text, handling images is more expensive regarding resources of the handheld device. Especially if a large number of graphics must be displayed at once, the relatively small screen size is one of the major drawbacks. The problem is also related to the provided screen resolution, which is lower than in stationary gadgets. Together, these lead to less graphical data which can be displayed. Moreover, an image is formed by many pixels that are discretely distributed. Special techniques are required to present and manipulate a large number of graphical contents on handheld devices. The other bottleneck is lack of processing power. These heavily affect the processing time, and therefore, usability of graphical data during presentation and interaction.

10.1 Icons as Representations of the World

An icon is a visual representation used to symbolize an object, action or relation. In the context of computing, an icon is used to refer to an image or graphic representation that suggests the purpose of an available function. In a visual language-based interface, it serves as alternative to text as a mode of communication, comprehension and expression. The icons represent the world. For example, `Fire`, `Fireman`, `Firetruck` and `PutFire` are necessary concepts that are represented in the form of icons for such an interface in the field of crisis management.

To date, there is not any standard icon database that can satisfy all requirements of a visual language-based application. Most applications develop their own icon database that is compatible for their visual language. Zlango, for example, includes icons in different grammatical units, such as nouns, pronouns, verbs, adjectives, and prepositions (Zlango, 2009). The language conveys simple everyday sentences and a direct translation of each icon in a sentence (Figure 10.1(b)).

In the field of crisis management, standard map symbols (Figure 10.1(a)) are promoted by the U.S. Government on a national basis for sharing information during emergency situations by emergency managers and people in responding to disasters (HSWG, 2003). It is based on research on guidelines and standards for the design of hazard and emergency maps and icons, including those of US military and NATO, by Dymon (2003).

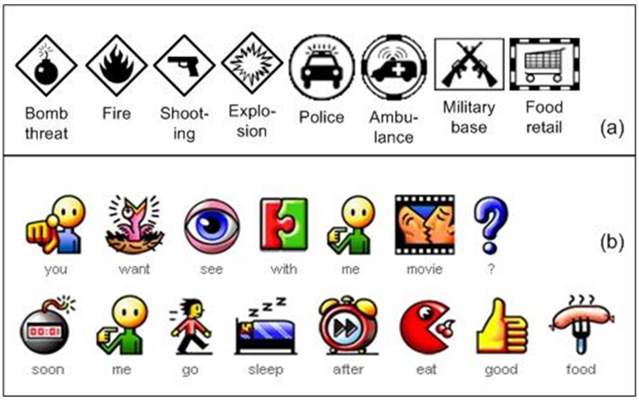


Figure 10.1: *Examples of (a) icons from HSWG (2003) and (b) sentences from Zlango*

Despite the obvious applicability of icons, poorly designed icons can have disadvantages, such as: (a) ambiguity occurs when an icon holds more than one meaning, (b) the meaning of an icon is assigned based on one's prior knowledge, (c) icons cannot always completely replace words, (d) design new icons that are interpreted properly by many users is costly, and (e) displaying many icons at once can make the user confused. A careful design, consistency and respect for conventions are all essential.

Table 10.1: *Requirements of icons for user interaction*

No.	Requirement on operational
1.	Give distinct appearance on a selected icon from the rest of unselected icons
2.	Whenever an icon is moved to a position overlapping another icon, but not in such a way as to activate any sensitive regions, the overlapping sensitive region of the moved icon shall be on top of the other icon
3.	Interacting with icons shall not destroy any user data without user permission
4.	Display a verbal label with each icon to assure that its meaning will be understood
5.	Group together related icons
6.	Give a clear visual discrimination between icons that represent available and unavailable functions
7.	Give a visual feedback that provides: (1) a visible indication to the user of the point of focus, that is the point at which the next user-selected action/interaction will occur and (2) a reference point so that the system can track where the icon is located

The use of computer icons is very strongly emphasized in various interface design guidelines and textbooks (Apple-Computer, Inc., 1992; Smith & Mosier, 1986; Shneiderman, 1986; Horton, 1994). A multi-part standard ISO/IEC 11581 (ISO/IEC, 2000) describes the graphical aspects and their associated functions necessary to allow a user to operate the GUIs of different software products. It provides a set of require-

Table 10.2: *Guidelines for designing icons*

No.	Guideline on design stage
1.	Represent the object or action in a familiar and recognizable manner
2.	Think about worldwide compatibility
3.	Avoid text in icons
4.	Make each icon distinctive from every other icon
5.	Use a consistent light source
6.	Optimize the icon design for the display on which it will most often been seen
7.	Maintain a consistent visual appearance in an icon family
8.	Use icon elements and colors consistently
9.	Use friendly and standard colors
10.	All available icons should be comprehensible. When first-time comprehension is not a usability requirement, then icons should be learnable and discriminable

ments and recommendations to enable the development and design of different types of icons to represents objects and functions on computer screens (Table 10.1). Most of these guidelines are basically mutually consistent to each other (Tabel 10.2).

10.2 Device-Oriented Icon Design

Raster and vector data are two ways to describe graphical content. Raster graphics are used in areas where a content description by geometrical objects is difficult or even impossible, such as in digital imagery. The image content of raster graphics is described by discrete image points (pixels) on a regular 2D-grid of a certain image size and precision. A common sizes of icons in most GUI-based software are 16x16 pixel and 32x32 pixel, high resolution program can support the maximum size up to 64x64 pixel. The ideal case is under different resolutions should have a set of corresponding big icons and small icons. Every pixel is independent from each other regarding its color, which causes a quite large file size. Thus, raster data is often stored in compressed representation, such as in GIF- or JPEG-format, only sometimes uncompressed, such as in BMP-format.

Vector graphics use geometric primitives and their attributes to describe image content. Such primitives are for instance points, lines, rectangles or circles, with attributes, such as fill-color, stroke-color or stroke-width. Graphical content described by vector primitives is in general smaller than raster data depending on the number and complexity of primitives. Simple graphics, such as icons, can be described by around 100 vector primitives.

In displaying graphics, data passes a number of steps (Figure 10.2). The first is loading the graphical content to memory, which is influenced by properties, such as file size and format. Since the content is often encoded, loading means file reading and decoding in memory. The decoding step transfers the content of a file to an internal memory representation (IMR), which form the basis for later display. The IMR is influenced by the properties image size and precision and the actual image content. The display step shows either the whole or portion of the IMR on display. A transformation function is used to map graphical content from a logical coordinate system to the physical display coordinate system. The graphical content user interactions is the main aspect that determined kind and properties of the transformation function.

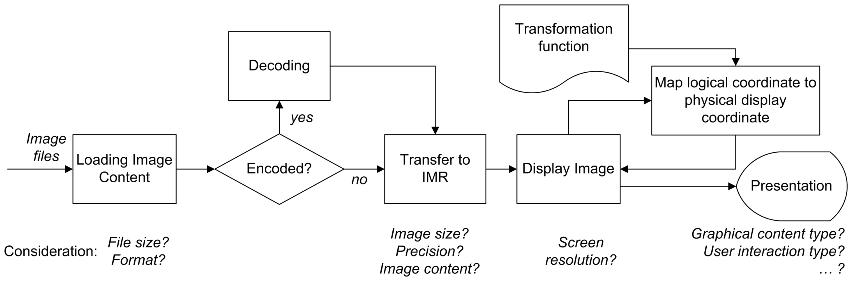


Figure 10.2: *Image display pipeline*

Rosenbaum et al. (2004) pointed out two main requirements for exploring a large number of graphics: (1) high presentation quality and (2) short presentation and update time. Since interaction is essential, performance is influenced by the used interaction technique, such as elementary zoom and pan. Interaction further requires the distinction in presentation, which spreads from loading until displaying the content, and update time, which considers the time to display the IMR. Small compressed files of raster graphics need more processing power for decoding, but are faster to read. However, the structure of the resulting IMR is mostly a bitmap and no conversion is necessary to map the IMR to display since both are based on the raster approach.

Modifications of raster graphics might be imposed by the transformation function, which influences the presentation quality. The most obvious approach for vector graphics is to store a description of the vector primitives as IMR, and to render them directly to screen at display time (direct drawing). A transformation matrix for modification of a vector graphic is preliminarily applied to primitives. Since no information gets lost during the transformation this delivers very good visual results, but costly processing power. It can be useful to render the vector graphic after decoding to an IMR and to discard the primitives (indirect drawing).

10.3 User-Oriented Icon Design

A workshop has been performed to review icon designs (Fitrianie, 2004). It was aimed at recognizing problems in introducing a new icon to users of a visual language-based interface using two research questions: how to represent a certain concept using an icon? and how to group icons so that they are easy to find?

Methodology

The workshop consists of the following three parts.

1. *Icon design test.* This part of the workshop was designed based on the icon evaluation in Equation 6.1. The participants were asked to recognize 100 icons from various concepts by selecting at least one meaning of every icon from five possible meanings. The participants could also freely suggest their own opinion for the answer.

2. *Icon grouping test.* In this part, the participants were asked to select one or more icons from six icons, that did not belong to the group. This test used 100 test cases.

3. *Interview.* Finally in this part, some questions were asked to the participants to review the entire test. The questions were mainly about their experience during the test and the design of the icons.

Eight people took part in the tests. Participants were selected in the range age of 25-50 years old. Most participants had worked with the computer for more than 30 hours per week. The level of English of the participants is relatively good although most of them are not native speaker. There was no assumption on distinction of gender or cultural differences.

The results of each part of the tests were processed separately. The answers were matched against the actual answers given by the designer to check the degree of user understanding of every icon. The icons, then, were grouped into three conditions: (1) "understand" if and only if all participants agreed with the designer, (2) "ambiguous" if at least one participant disagreed with the designer or gave more than one possible answer, and (3) "not understand" if at least one participant did not know the answer or gave a wrong one. The "ambiguous" and "not understand" icons were analyzed further to understand the reason.

Result Analysis

The following guidelines for designing icons are the most important points that came out from this study.

- *Use as many iconic and indexical icons as possible.* Iconic icons and indexical icons were easier to recognize than symbolic icons. In the first part of the workshop, from twelve only one iconic icon could not be recognized by our participants. Around 86% indexical icons were well recognized. However, from twenty-four symbolic icons only 50% were well recognized. Most of these icons represented abstract objects, prepositions, and verbs. Most participants mentioned that they had never seen most presented symbolic icons.

- *Relate an icon's meaning to a group's meaning.* In recognizing a new icon (Figure 10.3(a)), the test users made a guess about the individual icon based on some graphical features, such as elements, the color and the position of each element, and movement cues. When an icon belonged to a group of icons with a predefined concept (context), the test users tried to relate the icon's meaning to the concept or to other icons in the group. If these users did not understand the concept, they would abandon it and use the most probable initial guess.

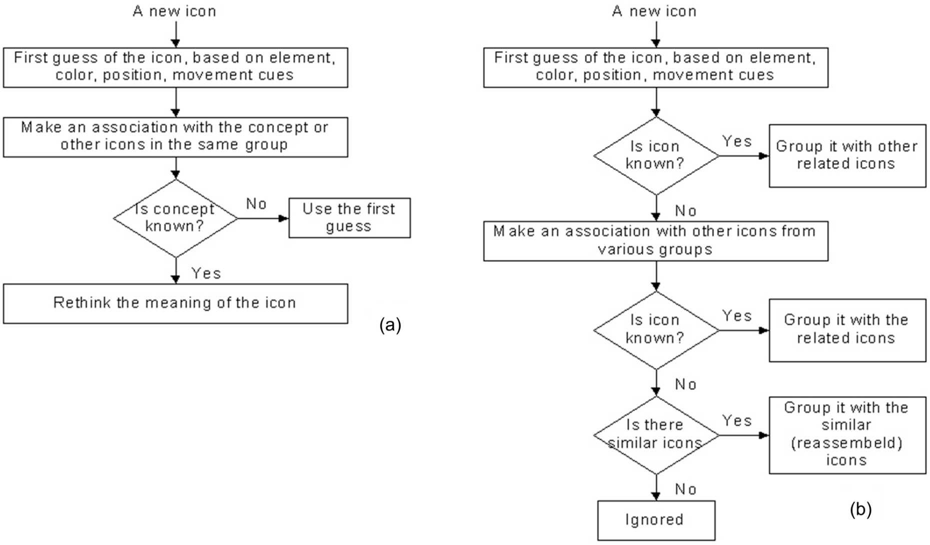

A new icon
First guess of the icon, based on element, color, position, movement cues
Make an association with the concept or other icons in the same group
Is concept known?
No
Use the first guess
Yes
Rethink the meaning of the icon
(a)
A new icon
First guess of the icon, based on element, color, position, movement cues
Is icon known?
Yes
Group it with other related icons
No
Make an association with other icons from various groups
Is icon known?
Yes
Group it with the related icons
No
Is there similar icons
Yes
Group it with the similar (reassembeld) icons
No
Ignored
(b)

Figure 10.3: *The flowchart of the user's thinking process in (a) recognizing an icon based on a concept and (b) grouping an icon around a concept (Fitrianie, 2004)*

In grouping an icon into a concept, the test users also made an initial guess (Figure 10.3(b)). If the icon's meaning was still unknown, they would try to associate it with other icons. Most of the time, this technique could reveal the meaning of the icon. Otherwise, the users would try to find icons that had similar elements and group the icon with these icons without knowing the meaning. If there was not any similar icon, the icon would be ignored. As pointed by Horton (1994), users tend to associate an icon with the rest of its group members to understand its meaning. Creation of an understandable concept of a group of icons leads to understandable icons. If the concept is unknown, there is a high chance the user will not recognize the icon.

- *Group icons based on their categories.* All participants grouped the icons based on the same category of objects. For example, "house", "school" and "shop" were in the "building" group; "not", "question" and "exclamation" were in the "notation" group, and "surprise" and "angry" were in the "emotion" group. It was advised to group indexical icons apart from iconic icons, for example the icon "scissor" both as a noun and as a verb ("cut").

- *Use movement visual cues for verb, proposition and adjective icons.* The participants miss-recognized most indexical icons that represented verbs, adjectives and prepositions. They could recognized some of them due to some cues elements, such as lines (for example, dashing effect for running), arrows (for showing direction such as "go up" or "go down"), words that represents human sounds (for example, "zzz.." for "sleep") and hand gestures (for transfer action, such as "give" or "take"). The proposition and adjective icons were mostly used to emphasized the meaning of other icons.

- *Careful use of popular icons.* Popular icons, such as road signs, software interface icons and institution icons, were easy to recognize. It was advisable to not alter the meaning of these popular icons. Moreover, icons that resembled or mimic real world objects might be recognized by users because of their prior knowledge about these objects. The results endorsed the research of Goonetilleke et al. (2001), which pointed out that trained users have shorter response time in recognizing icons than untrained users. In addition, the research also showed that a simple and short training could result in a considerable improvement in performance.

- *Design an image as simple as possible.* The meaning of an icons whose an image with a small number of elements were easier and quicker to recognize. Specific elements might create specific meaning. For example, using gestures for transfer act, such as "give" or "take".

10.4 Icons for A Visual Language-based Interface

To support the development of a visual language-based interface, in this research, we have developed a set of icons in the field of crisis management. The icons were distinguished for sentence constructions and those for operational purposes. Most of the design were not entirely new. They were designed based on or adopted from popular icons.

110 icons have been developed for constructing visual language-based messages. They were mainly designed based on selected crisis scenarios (see Appendix A.1). 75% designs of these icons were based on popular icons and some available databases, such as Zlango and HSWG. The design of an icon was mainly based on the literal meaning of the corresponding concept that is represented by the icon. The frame of an icon has a particular meaning that shows the type of the icon (Figure 10.4). The

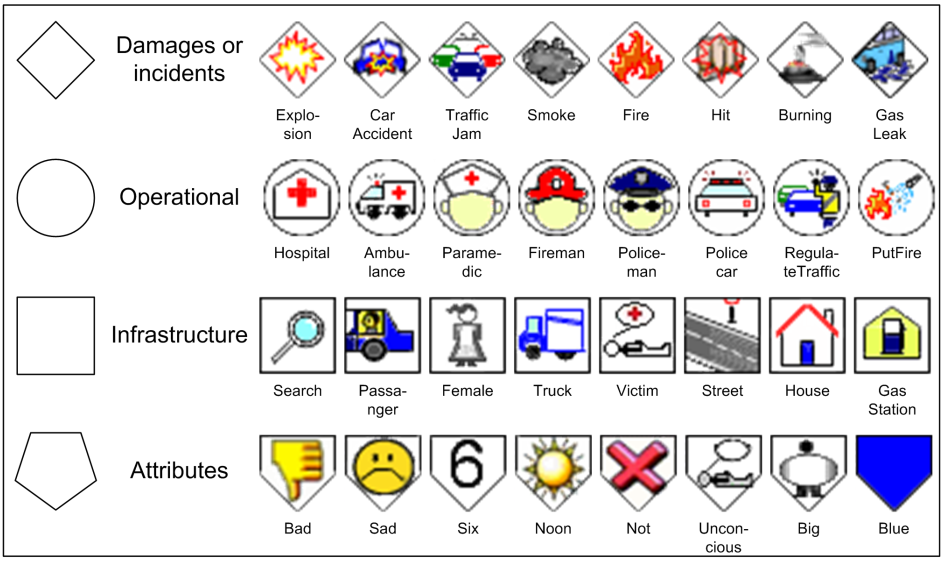
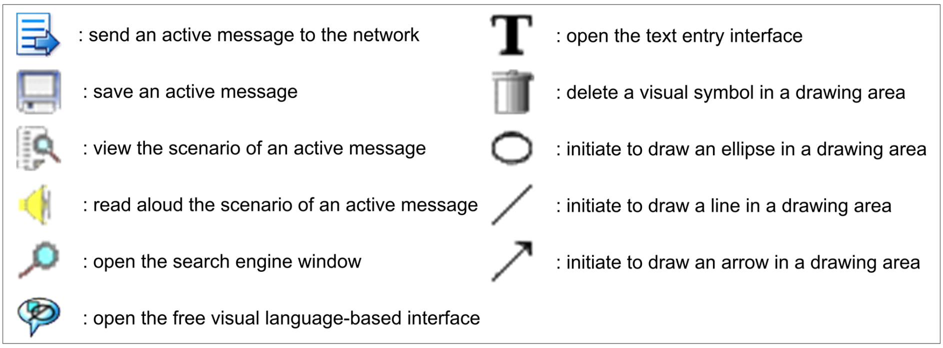


Figure 10.4: *Examples of developed icons*

icons were grouped into some concepts. An XML file links heading icons with icons in the same group. An icon can be grouped in more than one concepts.

The operational icons consist of two types: (1) the heading of icon groups, and (2) toolbar icons. The heading of a group is usually taken from the icons within the group itself. Ten toolbar icons have been developed (Figure 10.5). Nine icons were developed based popular icons. The new designs resemblanced the metaphor that represents the corresponding functionality. The size of these toolbar icons is 16x16 pixel. They have transparent background and do not have any frame.


: send an active message to the network

: save an active message

: view the scenario of an active message

: read aloud the scenario of an active message

: open the search engine window

: open the free visual language-based interface

: open the text entry interface

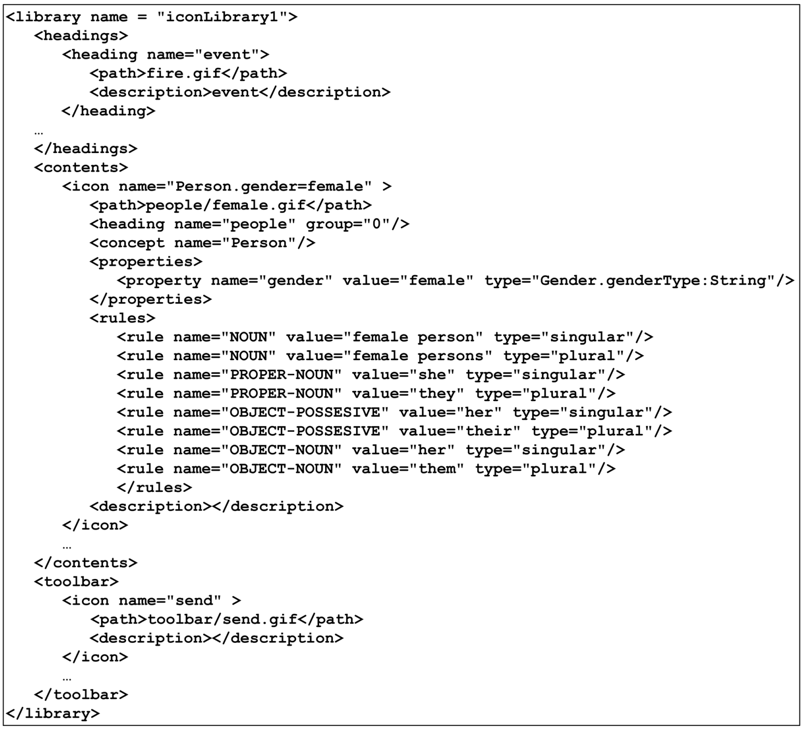
: delete a visual symbol in a drawing area

: initiate to draw an ellipse in a drawing area

: initiate to draw a line in a drawing area

: initiate to draw an arrow in a drawing area

Figure 10.5: *Toolbar icons and their descriptions*



```
<library name = "iconLibrary1">
   <headings>
      <heading name="event">
         <path>fire.gif</path>
         <description>event</description>
      </heading>
   …
   </headings>
   <contents>
      <icon name="Person.gender=female" >
         <path>people/female.gif</path>
         <heading name="people" group="0"/>
         <concept name="Person"/>
         <properties>
            <property name="gender" value="female" type="Gender.genderType:String"/>
         </properties>
         <rules>
            <rule name="NOUN" value="female person" type="singular"/>
            <rule name="NOUN" value="female persons" type="plural"/>
            <rule name="PROPER-NOUN" value="she" type="singular"/>
            <rule name="PROPER-NOUN" value="they" type="plural"/>
            <rule name="OBJECT-POSSESIVE" value="her" type="singular"/>
            <rule name="OBJECT-POSSESIVE" value="their" type="plural"/>
            <rule name="OBJECT-NOUN" value="her" type="singular"/>
            <rule name="OBJECT-NOUN" value="them" type="plural"/>
            </rules>
         <description></description>
      </icon>
      …
   </contents>
   <toolbar>
      <icon name="send" >
         <path>toolbar/send.gif</path>
         <description></description>
      </icon>
      …
   </toolbar>
</library>
```

Figure 10.6: *A part of the notation of the icon library (in XML format)*

10.5 System Design

Figure 10.7 presents the class diagram of the iconlibrary module. The module is aggregated in the Library class. The IconLibrary class contains a set of Icon objects and a set of Heading objects. A Heading object has a link to a list of Icon objects.

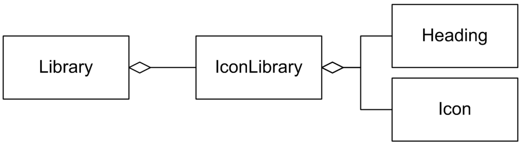


Figure 10.7: *The class diagram of the icon database*

The IconLibrary class receives an XML file that stores the notation of all icons and their headings (Figure 10.6). This class reads the file and extracts the contents into a list of Icon objects and a list of Heading objects. The message construction icons have a direct link to concepts in the ontology. This link is defined in the XML file too. When an icon is selected to construct a message, the corresponding concept of this icon is instantiated and its corresponding properties are filled in. The `Icon` class also stores information about how to convert an icon into its textual representation (`rules`). The conversion depends on which part of speech (POS) of the icon belongs to in the sentence. In addition, for a noun and a verb, this conversion depends also on the cardinality of the POS (plural or singular). Section 13.3 will elaborate on this conversion process.

Chapter 11
Emotion Analysis

In which research in two-dimensional affective lexicon database, human emotion expressions by analyzing cartoon illustrations and a method to analyze user emotional orientation is described.

Humans show verbal emotion by the choice of words. The emotional meaning determines the "effects" of words, especially the emotional tone effect on the interpretation of the speech content. Such effect is a meaningful marker and an occasional mediator of humans' mental, social, and even physical state (Clore, 1992). Besides informing one's emotional state, the use of such words is also the bridge to reality (Ricoeur, 1976). The way one describes events can define the meanings of the events, which help the other to improve context awareness.

11.1 Text-based Emotional Analysis

Text-based emotion analysis has been approached mostly as a text classification problem. A textual unit of certain sizes is classified as expressing positive or negative (or pleasant and unpleasant) feelings - *sentiment*. The unit size can go from words (Hatzivassiloglou & McKeown, 1997; Wiebe, 2000; Turney & Littman, 2003), phrases and sentences (Kim & Hovy, 2006; Wilson et al., 2005), and full texts (Hu & Liu, 2004). Previous approaches for assessing sentiment from text are based on one or a combination of the following techniques: keyword spotting (Turney, 2002; Pang & Lee, 2004), lexical affinity (Valitutti et al., 2004; Kim & Hovy, 2005), statistical methods (Pennebaker et al., 2003), using a dictionary of affective concepts and lexicon (Subasic & Huettner, 2001; Das & Chen, 2007), domain specific classification (Nasukawa & Yi, 2003), valence assignment (Polanyi & Zaenen, 2005; Shaikh et al., 2008), fuzzy logic (Subasic & Huettner, 2001), using topic knowledge (Mullen & Collier, 2004), and using the world knowledge (Liu et al., 2003).

Keyword spotting is the most simple approach among all text-based emotion analysis methods. Text is classified into emotional categories based on the presence of fairly unambiguous emotion words. Restriction to keywords is a suitable compromise if the input cannot always be clean and correct grammatical sentences. This approach relies on the presence of obvious emotion words. Using this method, a large-scale affective lexicon resource database is necessary for analyzing the emotional aspect of language. Elliot (1992) distinguished 198 emotion keywords (such as "happy", "sad", "angry") with their intensity modifiers (such as extremely, somewhat, and mildly) and cue phrases (such as "wanted to", "feel"). The affective lexicon from Ortony et al. (1988) was utilized to provide emotion words grouped into emotion categories. Other resources are an adjective database (Hatzivassiloglou & McKeown, 1997), an emotion class-based database (Ortony et al., 1988), an emotion expression database (Desmet, 2002), a common sense database (Liu et al., 2003) and an affective-WordNet (Strapparava & Valitutti, 2004). The largest database to date is created by Whissell (1989) - DAL, which contains 8742 words in a 2D circumplex model.

11.2 Are Emotion Words Really Bipolar?

The traditional notion of word meaning used in NLP is literal or lexical meaning as used in dictionaries. Adding to this notion, some researchers have brought subjective meanings of the degree of pleasureness (valence) and the degree of activation (arousal) specifically for words that express emotions. The arousal dimension tends to refer to the overall excitement or activation of the emotion, while the valence dimension tends to refer to how pleasing (positive) or displeasing (negative) the emotion is. Both degrees are depicted in a 2D space of pleasureness and activation. The qualitative values of emotion words is analyzed based on the social value that was learnt during interaction with others or objects in everyday life. The dimension displays the structure of emotions in a comprehensible way. The major criticism of this approach is that, although it can categorize emotions, the approach is not sufficient to differentiate between emotions. For example, the emotion "annoyed" and "fearful" (in Figure 11.1(a)) or "indignant" and "alarmed" ((in Figure 11.1(b)) fall close together on the circumplex. On the other hand, for an automatic reasoning system, quantitative data is more efficient.

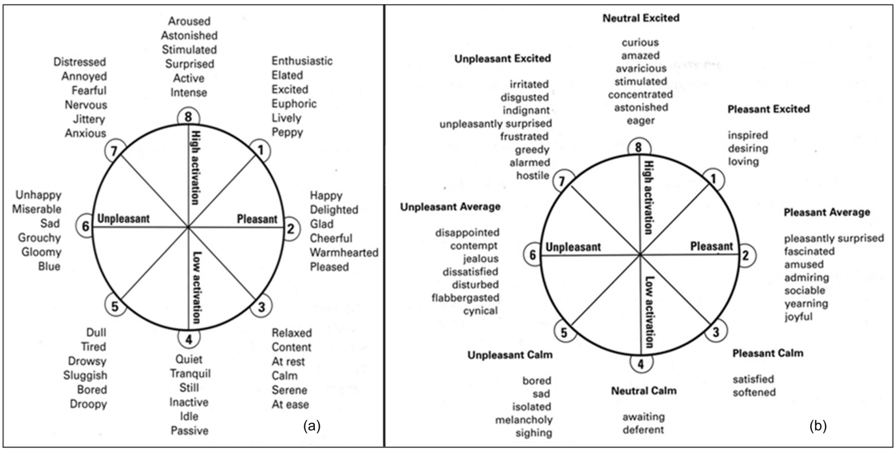


Figure 11.1: *The circumplex of emotions (the numbers represent octants): (a) adapted from Russell (1980) and Waston & Tellegen (1985) and (b) from Desmet (2002)*

Kamps & Marx (2002) brought other notions of meaning into NLP. They proposed the differences between the relatively objective notion of lexical meaning and more subjective notions of emotive meaning by exploiting WordNet. The basic notion of the main synonymy (synset) relation of WordNet is denoting coincidence of lexical meaning (Fellbaum, 1998). Based on this, the research related emotion words with the following rules:

$$MPL(w_i, w_j) = n \tag{11.1}$$

where n is the smallest number of synset steps between two words. For example, $MPL(good, bad) = 4$ {good, sound, heavy, big, bad}.

$$TRI(w_i, w_j, w_k) = \frac{MPL(w_i, w_k) - MPL(w_i, w_j)}{MPL(w_k, w_j)}, w_j \neq w_k \tag{11.2}$$

where $TRI(wi, wj, wk)$ is the relative distance of a word to two reference words. If this equation is applied for two reference words, this will derive:

$$EVA^*(w_i) = TRI[w_i, w_j, w_k) \tag{11.3}$$

where w_j and w_k are constant. The value of EVA^* allows us to distinguish between words that are predominantly used for expressing positive emotional states (close to 1), for negative emotional states (close to -1), or for non-emotion words (around 0).

An experiment has been performed to study the subjective meaning of words especially used in dialogues. We assumed that this meaning could evoke emotion expressions, such as facial expressions. This was based on considerations that: (1) the human face is the primary channel to express emotion, (2) the instantaneous emotional state is directly linked to the displayed facial expressions (Ekman, 1999) and (3) the expressions occur synchronously to one's own speech or to the speech of others (Ekman, 1979; Condon & Osgton, 1971). Moreover, Tokuhisa et al. (2006) pointed out that the accuracy of text-based emotion annotation was twice higher when corresponding facial expressions were referred. For the purpose of this experiment, a video recording of dialogues between pairs of participants from Wojdel (2005) was utilized. These participants were requested to perform dialogues about different topics and show as many expressions as possible

Methodology

We collected words that have a direct link to the facial expressions. Firstly, three independent observers marked the onset and offset of an expression. Secondly, the expressions were labeled according to the context. The agreement rates between the observers in both steps were about 73%. Finally, the emotion words used in each expression were collected. The dialogue (manually) was partitioned into basic constituents (a single sentence) and for each constituent into components (single words and punctuation). Then, for each component, the time of its occurrence (number of frames in which the given word is pronounced) was determined. The next, for each selected facial expression, the text that started with the component synchronized with the first frame of a given facial expression and ends with the component synchronized with the last frame of this expression, also was determined.

Furthermore, two approaches were used to plot the quality and intensity of both labels and emotion words, which are found in the experiment, into a 2D space. In the first approach, the EVA^* function (eq. 11.3) is utilized to develop a 2D space with bipolar dimension of valence and arousal. In the second approach, multidimensional scaling (MDS) represents emotion words, also in 2D space, using information relative to "similarity" (corresponding meaning) between each couple of emotion words.

Result Analysis

The experimental results indicated that the participants showed most of the time a neutral face. However, in total 40 different facial expressions (about 20-35 different expressions per participant in each dialogue) and 140 emotion words were captured. We related the labels of these facial expressions to the emotion expressions labels used by Desmet (2002) (see Figure 11.1(b)). The experimental results showed that most facial expressions (around 63%) corresponded to the text spoken. 54.6% of the emotion words spoken by the participants corresponded to facial expressions (Table 11.1). From this experiment, it was not possible to draw a direct link between emotion words and facial expressions due to some words that occurred too sparse and ambiguous (in different context they were related to different expressions).

Table 11.1: *Statistics of words in the input text linked to facial expression*

Words	Total	Linked to Expression
Non-emotion words	2206	1022 (46.3%)
Emotion words	140	77 (55%)

Figure 11.2 shows 140 emotion words plotted in 2D space pleasant-unpleasant and active-passive using the relative distance rules. Here, the relatedness values EVA^* for some emotion words were not as expected. For example, the valence score for $EVA^*(joyful)$ returns negative while for $EVA^*(angry)$ returns positive. On the other hand, the arousal score for $EVA^*(sad)$ returns positive while for $EVA^*(dramatic)$ returns negative.

Figure 11.3 shows selected 38 emotion words plotted based on the MDS mapping. The procedure found the configuration or cluster that approximated the observed distances in the best way. Initially, the MPL function (eq. 11.1) was used to construct an N x N input matrix. The Euclidian distances among all pairs of points were applied to measure natural distances of those points in the space. By lowering the degree of correspondence between the Euclidian distance among points and the input matrix, the best corresponding MDS map were achieved. Six dash curves indicate clusters of

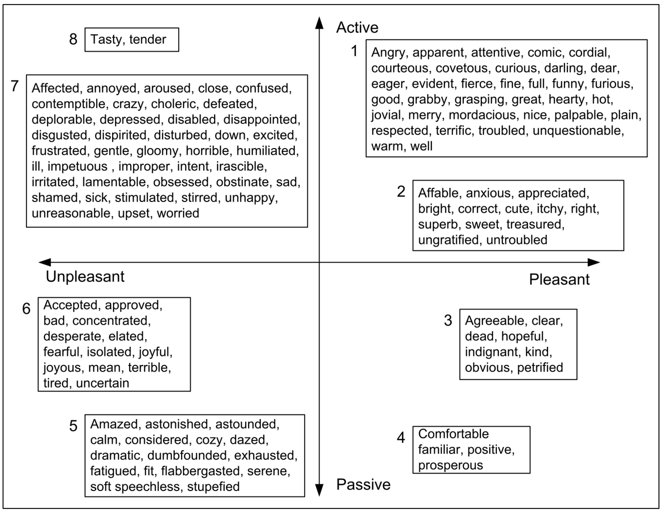

Active
Passive
Unpleasant
Pleasant
1 Angry, apparent, attentive, comic, cordial, courteous, covetous, curious, darling, dear, eager, evident, fierce, fine, full, funny, furious, good, grabby, grasping, great, hearty, hot, jovial, merry, mordacious, nice, palpable, plain, respected, terrific, troubled, unquestionable, warm, well
2 Affable, anxious, appreciated, bright, correct, cute, itchy, right, superb, sweet, treasured, ungratified, untroubled
3 Agreeable, clear, dead, hopeful, indignant, kind, obvious, petrified
4 Comfortable familiar, positive, prosperous
5 Amazed, astonished, astounded, calm, considered, cozy, dazed, dramatic, dumbfounded, exhausted, fatigued, fit, flabbergasted, serene, soft speechless, stupefied
6 Accepted, approved, bad, concentrated, desperate, elated, fearful, isolated, joyful, joyous, mean, terrible, tired, uncertain
7 Affected, annoyed, aroused, close, confused, contemptible, crazy, choleric, defeated, deplorable, depressed, disabled, disappointed, disgusted, dispirited, disturbed, down, excited, frustrated, gentle, gloomy, horrible, humiliated, ill, impetuous , improper, intent, irascible, irritated, lamentable, obsessed, obstinate, sad, shamed, sick, stimulated, stirred, unhappy, unreasonable, upset, worried
8 Tasty, tender

Figure 11.2: *2D space-emotion words in octants-related pleasantness-activation*

words, which are clustered due to the similarity of meaning between the words. As it was expected, some words could not be clustered, such as "indignant" is separated from the group of "fierce", "angry", and "furious".

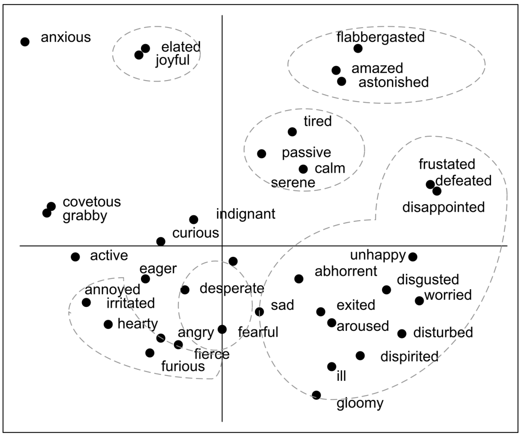


Figure 11.3: *MDS-emotion words mapping*

Both approaches may not be accurate enough to define the degree of emotion words in a 2D pleasantness-activation. Table 11.2 shows the comparison of the results with other 2D affective lexicon databases. Kamps & Marx (2002) have already mentioned that the values obtained from the relatedness equation do not give a precise

Table 11.2: *Comparison the experiment results to other 2D affective lexicon databases*

Affective Lexicon Database	#Corresponding Words	Degree of Compliance
Whissell (1989)	140	80.3%
Desmet (2002)	21	71.4%
Russell (1980)	22	31.8%

scale for measuring degrees of emotion. However, a weaker relation between words used to express sentiment and their distance can be derived from this approach. However, the results somehow endorse the emotion words classification based on subjective and psychological research.

11.3 Are Expressions Always Parallel to Emotions?

Emotion influences the choice of emotion expressions, such as the use of language, facial expressions, body gesture, prosody. In a dialogue the emotional state is co-determined by the events that happen during a dialogue. Many research showed that the capability of an agent communicating with humans using both verbal and nonverbal behaviors will make the interaction more human-like and intimate (Prendinger & Ishizuka, 2001; Schiano et al., 2000). To enable rich expressivity of this agent, the emotion displays should show a correct expression of the state of an agent in the dialogue. Directly displaying the emotion expressions from a dialogue-text, however, provides challenges on several levels.

Firstly, emotional linguistic content consists of entities of complexity and ambiguity, such as syntax, semantics and emotions. Since words are interrelated and influence each other's affect-related interpretation, a model that only analyzes emotive meanings of words used within a phrase or a sentence may fail to describe emotion expressed by a complete phrase or sentence. Secondly, along with the role emotional signal, other functions of human emotion displays should also be taken into account, such as: (a) communicative function, indicating the mental state of speaker and listener (for instance alignment, acknowledgement and the like) (Vark et al., 1996) and (b) interpersonal functions, showing how one feels/thinks about the other and the relation to one self (Mulder et al., 2004). Finally, the expressions for emotions are certainly not always trustful (Ekman, 2001). Emotion expressions do not always correspond to felt emotions: they can be faked (showing an expression of an unfelt emotion), masked (masking a felt emotion by an unfelt emotion), superposed (showing a mixed of emotions), inhibited (masking an emotion expression with neutral expression), suppressed (de-intensifying an emotion expression) or exaggerated (intensifying an emotion expression).

A study in building knowledge on how to appropriately express emotions for text-based dialogue agents has been performed. The purpose of this study was to find out whether the emotion expressions are in correspondence (parallel) to the emotion loading of the dialogue-text. The previous study has shown that it was difficult to simulate facial expressions using predefined dialogues. Therefore, the following study analyzed the appearance of facial expressions and the corresponding dialogue using characters of selected cartoon illustrations. The cartoon illustration was chosen as data set because: (1) the emotion expressions in cartoons are associated with particular human emotions, (2) each character shows emotion with largely expressive facial displays; therefore the emotion can be observed easily and (3) the data includes the context why an emotion occurs and why a particular facial expression is displayed.

Methodology

Five cartoon illustrations consisting of 27 dialogue fragments (277 dialogues - 460 facial expressions and 357 text balloons) were selected, which contained dialogues between two characters in diverse situations that evoked various emotional states. Nine groups of two students were asked to annotate the data. One member of every couple only annotated the facial expressions independently from the other member, who annotated only the dialogue-text. The experiment was performed as follows:

- The participants with the facial expression corpus were asked to: (1) identify emotion expressions, (2) label the identified expressions using emotion words and (3) identify the function of the expression in the dialogue.

- The participants with the dialogue-text data were asked to: (1) collect emotion words in the dialogue-text, (2) label the dialogue text using emotion words and (3) identify the function of the expression in the dialogue.

The communication function category consists of question, acknowledgement, statement, clarification and confirmation (Vark et al., 1996).

Result Analysis

The agreement rates between raters of the facial expressions were 78% and of the dialogue-text were 73%. Table 11.3 shows that the participants used a large range of emotion words (54 words) to label the emotion types of the characters in the stories: 43 labels from the facial expressions and 34 labels from the text corpus. The coherent results were formulated by grouping the emotion words into eight octants based on Desmet (2002). In these groups, joyful (18%), angry (14%), unpleasantly-surprised (13%) and (pleasantly-) surprised (12%) werw shown most often.

To study which emotion words correlated to a certain emotion, 207 dialogue-lines (consisting of 357 dialogue-texts) were selected, in which the characters showed their facial expression and prompt text. About 23% of the dialogue-text contained explicit emotion words, such as "great" and "bad" (adjective) or "hate" and "love" (verb). Other dialogues (about 41%) needed our common sense to correlate them with a certain emotion, such as "how funny! It really works" for amused and "this time he's

Table 11.3: *54 emotion labels found in the corpus (in eight octants of Desmet (2002))*

No.	Valence-Arousal	Facial Expression Labels	Dialogue-Text Labels
1.	Pleasant-Excited	{inspired, 5}, {desiring, 3}, {in love, 4}	{in love, 4}, {inspired, 5}
2.	Pleasant-Average	{surprised, 57}, {happy/ cheerful, 83}, {arrogant/ amused, 15}, {teasing/ admiring, 11}	{teasing, 3}, {surprised, 31}, {cocky/ conceited/ amused, 15}, {good-humored/ delighted/ cheerful, 77}
3.	Pleasant-Calm	{satisfied/pleased, 5}	{militant/pleased, 9}
4.	Neutral-Calm	{hope/calm/normal, 23}	{neutral/hope, 61}
5.	Unpleasant-Calm	{sighing, 2}, {bored, 2}, {sad/distressed, 23}	{sighing, 2}, {sad, 11}
6.	Unpleasant-Average	{jealous, 5}, {disappointed, 9}, {wicked/ mischief, 27}, {confused/ dumbfounded/ scared, 17}	{wicked/ mischief/ roguish, 16}, {confused/ fear/ bewildered, 9}
7.	Unpleasant-Excited	{alarmed, 3}, {hostile, 2}, {greedy, 2}, {hate/ disgusted, 11}, {irritated/ disturbed, 11}, {pleading/ frustrated, 1}, {angry/ taken-back, 66}, {shocked/ unpleasant-surprised, 59}	{shocked, 31}, {hostile, 1}, {annoyed, 5}, {greedy, 4}, {desperate/ pleading, 3}, {angry/ fed-up, 49}
8.	Neutral-Excited	{curious, 2}, {amazed, 7}, {concentrated/ confident, 7}, {eager/ excited, 11}	{enthusiastic, 18}, {amazed, 3}

going flat on his beak!" for indignant. Textual-prosody were also used to show the character's emotion (about 16%), such as "grrrr" (indignant), "gloek" (flabbergasted), and "snap!" (hostile). The rests (about 19%) did not contain any emotion words and were considered as neutral, while the facial expressions showed a certain emotion.

The emotion labels of the facial expressions did not always correspond with the dialogue-text. As shown in Table 11.4, negative emotions were more often masked or inhibited, while the positive emotions were always conveyed. In the story, the characters masked their negative emotions with positive emotions to pretend their actual feeling (usually bad intention to others). A fake joyful emotion was often used to mask negative emotions. The characters suppressed or even inhibited their negative emotions if they needed to be polite or if they were feeling shameful. Only 1% of neutral emotions were expressed as neutral. The characters used facial displays to complete or even (in some cases) exaggerated the expression of an emotion. Superposed emotions were shown for surprised with joyful or anger, however, most annotators labeled them as surprised and shocked.

Table 11.4: *The correspondence of 357 dialogue-text and facial expressions*

Octant#		Facial expression corpus							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Dialogue-text corpus	**1**	9							
	2		126						
	3	1	3	5					
	4	5	19	6	7		6	13	3
	5					12	1		
	6				2		25		
	7		7	1	3		13	69	
	8		10						11

Emotion words that were used in 83 dialogue-texts were collected to study whether the facial expressions correlated with the use of emotion words in the text. For each character i and each dialogue-text j, the words $w_{1..n}$ were plotted on two-dimensional "arousal" and "valence" using the following equation:

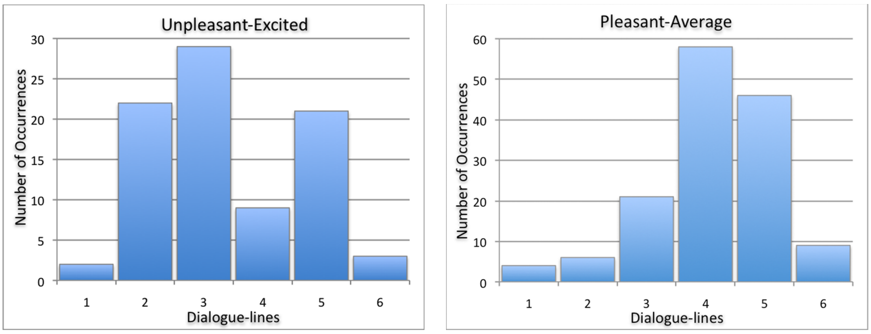
$$v_{ij} = 1/n\left(v_{w_1} + v_{w_2} + \ldots + v_{w_n}\right); a_{ij} = 1/n\left(a_{w_1} + a_{w_2} + \ldots + a_{w_n}\right)$$

The valence $v\,[-1\ldots1]$ and arousal $a\,[-1\ldots1]$ scores of a word w were obtained from DAL Whissell (1989). The equation measured the text emotional-orientation. Then, the circumplex were divided into the eight octants using the following rules: (1) $-0.2 \leq v \leq 0.2$ is neutral, $v > 0.2$ is pleasant, $v < -0.2$ is unpleasant and (2) $-0.2 \leq a \leq 0.2$ is average, $a > 0.2$ is excited, $v < -0.2$ is calm. Finally, each character and each dialogue line were mapped with the corresponding facial expression corpus to find whether they corresponded to each other. Table 11.5 shows almost similar composition as table 11.4. This indicated that the characters showed their felt emotion in most cases. Some emotion words were still categorized into different octants from the results done by raters. This could be means that the characters masked or inhibited their emotions.

Table 11.5: *The correspondence of 83 dialogue-text and facial expressions*

Octant#		Facial expression corpus							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Dialogue-text corpus	**1**	4							
	2		27						3
	3		1	1					
	4				1				
	5				2	5	1		
	6						8		
	7		3	1			3	14	2
	8		3						3

In the final experiment, the distribution of the elapsed frames of facial expressions was analyzed. For this purpose, appropriate histograms for each octant were plotted with interval one frame (one dialogue-line). From the histograms in, emotion expressions mostly appeared for 2-3 frames. However, 40% of unpleasant-excited and 90% of pleasant-average could appear until 4-6 frames (see Figure 11.4). The pleasant-average histogram also contained a tail comprising of the longest observed expression duration. This might be attributed to the dual role of the expression joyful and anger as both short emotion signal and mood indicator. This study also showed that the changing emotions between octants mostly occurs (92%) when the characters acknowledged a new event or object. The resulted emotion was directed or provoked by the event or object.



(a) (b)

Figure 11.4: *Histogram for elapsed time: (a) unpleasant-excited and (b) pleasant-average*

11.4 Emotion Orientation Analyzer

Emotional tone in a linguistic content can be expressed using specific keywords that represent important concepts in the focused domain. Such indicators can be in the forms of nouns, verbs and adjectives. The keywords have semantic orientation, such as positive or negative emotion and active or passive emotion. This semantic orientation may be diminished or even changed by three types of emotional shifters: negations (they reverse the polarity of a term), intensifiers and diminishers (they affect the degree to which a term is positive or negative).

In this research, a method for analyzing the way people use certain keywords to describe events was explored. The keywords may mark their emotional state and (at the same time) indicate the direction of the emotion. They can define the emotional meanings of the events and, as a result, can help to improve context awareness. Sperber & Wilson (1986) pointed out that as long as both speaker and audience share the same cognitive environment and facts, the communication of a given situation is successful. In such a condition, a certain indicator word in a language can have the potential to succinctly summarize a complex situation. In the case of a crisis management system that is potentially overloaded with flood of inputs, non-urgent and irrelevant reports, complaints, and unnecessary questions, such as "my window is broken", "people are running away in panic", "I am afraid" and "is it over yet?", cannot help crisis responders seeking information about what is going on. However, information, such as "we've just hit badly by a threatening earthquake", "I heard a nasty loud bang", and "a man is seriously injured", can describe the crisis event and hint the urgency level of the situation. Humans can easily distinguish non-urgent cases from urgent and emergent cases due to keywords, such as "earthquake", "threatening", "hit", "nasty", "seriously" and the like.

In the process of automatically classifying the emotion orientation, the proposed emotion analyzer works by recognizing keywords and their semantic scores provided by an affective lexicon database. The emotion analyzer works as follows (Figure 11.5): (1) identify important keywords, emotional shifter and aggressive words, (2) process the negation of the keywords by selecting their direct antonym, (3) calculate the diminished or intensified semantic scores of all identified keywords, (4) calculate the scores of the input, which is the average semantic scores of all identified keywords and (5) update the value of the emotion thermometers and the urgency level of the dialogue. Each component in the Figure 11.5 is described in the following subsections. The performance of the analyzer is indicated with some examples.

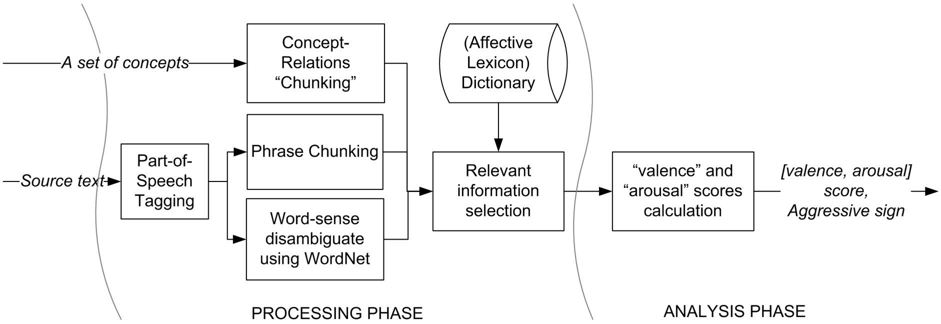

A set of concepts
Source text
Concept-Relations "Chunking"
Phrase Chunking
Part-of-Speech Tagging
Word-sense disambiguate using WordNet
(Affective Lexicon) Dictionary
Relevant information selection
"valence" and "arousal" scores calculation
[valence, arousal] score, Aggressive sign
PROCESSING PHASE
ANALYSIS PHASE

Figure 11.5: *Schematic view of the text-based emotion orientation analyzer*

11.4.1 Affective Lexicon Dictionary

This emotion analyzer uses four dictionaries:

1. ***Collection of nouns, verbs, adjectives and adverbs.*** In the current implementation, the scores defined in the DAL were applied (Figure 11.6). Each word has

a valence and arousal score in the interval [1..3]. For example, the valence and arousal score of "bad" is 1.2857 and 1.4615, respectively.

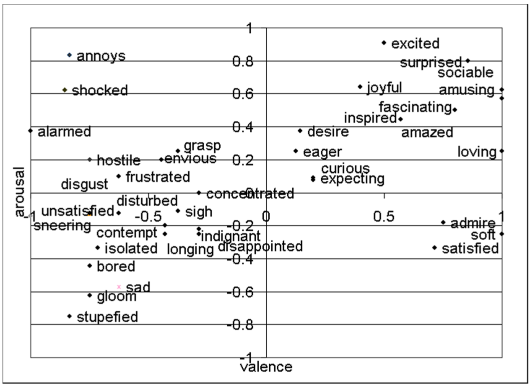


Figure 11.6: *38 adjectives from DAL depicted in 2D valence and arousal* $[-1..1]$

2. ***Collection of indicator keywords in the domain in focus.*** Keywords in the field of crisis management (200 words = 103 verbs, 27 adjectives and 70 nouns) were collected (see Chapter 9). More than 92% of these words were known in DAL.

3. ***Collection of 85 aggressive words.***

4. ***Collection of qualifiers.*** 28 simple qualifiers, such as "very", "rather", "huge", "low" and 32 compound qualifiers, such as "not very ..", "more or less ..", and 'sort of ..", were collected. They were assigned and ranked heuristically based on their intensity I_q [0..1] (from "not at all" to "very" and from "tiny" to "huge"). For example, $I_{very} = 1$, $I_{medium} = 0.5$, $I_{slightly} = 0.2$, and $I_{not\_at\_all} = 0$.

To assign their semantic scores of words that are out of DAL's domain, WordNet (Fellbaum, 1998) is used for finding the synonym and hypernym of these words.

11.4.2 Relevant Information Selection

In the processing phase, the analyzer extracts the input C in three steps. Firstly, for each text input, a parser in the analyzer uses a (n-grams) probabilistic tagger that returns the POS of each word (such as noun, verb, and adjective) in a sentence. If C is a set of concepts, then the analyzer takes all concepts and applies their relations. Let w can be a noun or a verb, $\mu = \{w_1, w_2, \ldots, w_n\}$ represents a conjugation of nouns or verbs, and e represents a word or a phrase (or a concept) that explains μ. The parser extracts and maps C onto set of phrases p. Each $p \in C$ is composed by a pair of $\{e, \mu\} \in C$, where $e = \{(q_1, \tau_1), \ldots (q_m, \tau_m)\}$ and $q \in C$ is one or more qualifiers that can intensify or diminish the absolute quality of the fact $\tau \in C$ (adjective or adverb). For example, for $C =$ "we're just attacked very badly by a quite threatening and long earthquake" can be defined into: $p_1 = ("we", null)$, $p_2 = ("attack", ("very", "badly"))$, and $p_3 = ("earthquake", (("quite", "threatening"), ("long", null))$.

Secondly, the parser discards phrases that do not contain any indicator keyword. Here, $C = \{p_2, p_3\}$ are processed since "attack" and "earthquake" are considered as indicator keywords. In addition, this parser also identifies aggressive words. Finally, for any negation word (such as "not", "no", "never"), the parser executes the following algorithm:

```
if the negation negates any $\mu_i$ in a certain $p_i$ then
   this $p_i$ is discarded from $C$
else if the negation negates any $\tau_{ij}$ in a certain $e_{ij}$ then
   this $\tau_{ij}$ will be replaced by its direct-antonym
   (based on WordNet) in the DAL
```

For example: the phrase "(is) not an earthquake" will be discarded from C because the emotional orientation of "earthquake" becomes zero. The phrase "not a severe earthquake" will be replaced by "a simple earthquake". The phrase "not very" in "not very long earthquake" will be treated as a qualifier and has an intensity value I_{not_very}. The parser assigns semantic scores for all words.

11.4.3 Word-Orientation Count Method

In the analyzing phase, the emotion analyzer calculates the semantic scores of C in the following five steps:

1. ***Convert the semantic scores into the interval*** [0..1]. The valence and arousal scores of the DAL correspond to the probability of a word being pleasant and active, respectively. For example, $v_{word} = 2$ means: (a) the word is neutral and (b) v_{word} is equivalent with probability value 0.5 since 50% raters rates it being positive, which implies that with the same probability the word is negative. Therefore:
$$\forall word_i inDAL, v'_{word_i} \in [0..1] = \left(v_{word_i} - 1\right)/2$$
$$\forall word_i inDAL, a'_{word_i} \in [0..1] = \left(a_{word_i} - 1\right)/2$$

2. ***Calculate the semantic scores of*** μ_i. The semantic scores of μ_i are calculated as the extreme scores of all w_{ij} from the largest number of w_{ij} with the same emotional orientation. For example, if more than 50% of w_{ij} has $score$ more than 0.5, the maximum score will be taken.
$$v_{\mu_{ij}} = EXTREME\_VALUE\left(v_{w_{i1}}, v_{w_{i2}}, \ldots, v_{w_{in}}\right)$$
$$a_{\mu_{ij}} = EXTREME\_VALUE\left(a_{w_{i1}}, a_{w_{i2}}, \ldots, a_{w_{in}}\right)$$

3. ***Calculate the semantic scores of*** e_i. All qualifiers are represented and operated in fuzzy theory. The I values ($\in$ [0..1]) forms a fuzzy set of q_{ij}. The inverse of S

function (Figure 11.7) is used to model the membership function of this fuzzy set, which treated the semantic scores as parameters to adjust the membership function. The new scores of τ_{ij} is assumed never be less than 0, since the negation of μ_i has been taken care in the processing phase. Therefore:

$$I_{q_{ij}} = S\left(v_{\tau_{ij}}; v_{\tau_{ij}} - \xi, v_{\tau_{ij}}, v_{\tau_{ij}} + \xi\right) = S\left(a_{\tau_{ij}}; a_{\tau_{ij}} - \xi, a_{\tau_{ij}}, a_{\tau_{ij}} + \xi\right)$$

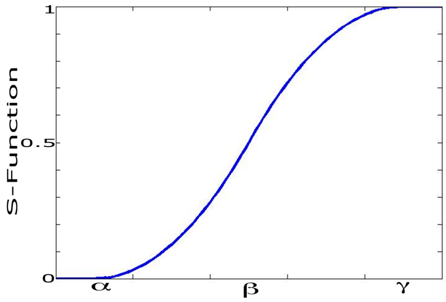
$$v'_{\tau_{ij}}\left(I_{q_{ij}}, v_{\tau_{ij}}, \xi\right) = S^{-1}\left(v_{\tau_{ij}}; v_{\tau_{ij}} - \xi, v_{\tau_{ij}}, v_{\tau_{ij}} + \xi\right)$$

$$v'_{\tau_{ij}} = 0 \quad \text{if } v'_{\tau_{ij}} \leq v_{\tau_{ij}} - \xi \quad \text{and } v'_{\tau_{ij}} = 1 \quad \text{if } v'_{\tau_{ij}} \geq v_{\tau_{ij}} + \xi$$

$$a'_{\tau_{ij}}\left(I_{q_{ij}}, a_{\tau_{ij}}, \xi\right) = S^{-1}\left(a_{\tau_{ij}}; a_{\tau_{ij}} - \xi, a_{\tau_{ij}}, a_{\tau_{ij}} + \xi\right)$$

$$a'_{\tau_{ij}} = 0 \quad \text{if } a'_{\tau_{ij}} \leq a_{\tau_{ij}} - \xi \quad \text{and } a'_{\tau_{ij}} = 1 \quad \text{if } a'_{\tau_{ij}} \geq a_{\tau_{ij}} + \xi$$

where $\xi = x - \alpha = \beta - x$ as an adjusted coefficient factor.



$$S = \begin{cases} 0 & \text{for } x \leq \alpha \\ 2\left(\frac{x-\alpha}{\gamma-\alpha}\right)^2 & \text{for } \alpha \leq x \leq \beta \\ 1 - 2\left(\frac{x-\gamma}{\gamma-\alpha}\right)^2 & \text{for } \beta \leq \text{x} \leq \gamma \\ 1 & \text{for } x \geq \gamma \end{cases}$$

Figure 11.7: $S - function(x; \alpha, \beta, \gamma)$

Here, four assumptions were taken: (a) $v_{\tau_{ij}}$ and $a_{\tau_{ij}}$ values always correspondence to $I = 0.5$, (b) $\xi = 0.5$, (c) if τ_{ij} is null, then $\tau_{ij} = \mu_{ij}$. For example, $p1 = \{"earthquakes", ("short", null)\}$ with I_{short}, and (d) if there are more than one τ_{ij} in C then this calculation works recursively. For example, in C ="the earthquake is very short"' both quantifiers "very" and "short" have different intensity value of I_{very} and I_{short}, therefore, the decomposition of C becomes $p_1 = \{earthquake, \{very, e_1\}\}$, where $e_1 = \{short, null\}$ and is calculated using the score of "earthquake" as the value of τ_{ij} (see the third assumption above). Then the semantic scores of e_i are:

$$v_{e_i} = EXTREME\_VALUE\left(v'_{\tau_{i1}}, v'_{\tau_{i2}}, \ldots, v'_{\tau_{in}}\right)$$

$$a_{e_i} = EXTREME\_VALUE\left(a'_{\tau_{i1}}, a'_{\tau_{i2}}, \ldots, a'_{\tau_{in}}\right)$$

4. ***Calculate the semantic scores of*** p_i. Given the independent μ_i and e_i being pleasant with probability v_{μ_i} and v_{e_i} being active with probability a_{μ_i} and a_{e_i}, respectively, the semantic scores of p_i are:

$$v_{p_i} = \left(v_{\mu_i} + v_{e_i}\right) - \left(v_{\mu_i} \bullet v_{e_i}\right); \quad a_{p_i} = \left(a_{\mu_i} + a_{e_i}\right) - \left(a_{\mu_i} \bullet a_{e_i}\right)$$

This calculation can be recursive. For example, for "the earthquake that shook the earth severely":
(1) $p_1 = \{"shook", (null, "severely")\}$, and
(2) $p_2 = \{"earthquake", (null, p_1)\}$.

5. ***Calculate the semantic scores of*** C. These scores are the mean scores of all collected phrases in C.

$$v_c = 1/n\left(v_{p_1}, v_{p_2}, \ldots, v_{p_n}\right); \quad a_c = 1/n\left(a_{p_1}, a_{p_2}, \ldots, a_{p_n}\right)$$

Table 11.6 shows examples of user inputs.

Table 11.6: *Examples of user inputs; The negative orientation:* $D > C > B, A > B, C \approx A$

No.	Input	Valence Score	Arousal Score
A	I saw a building collapsing.	-0.4286	0.1429
	Fire burst out.	-0.6364	0.667
	I just heard an explosion.	-0.778	1
B	A little fire occurs around the building.	-0.3504	0.6098
	Small explosions near the neighborhood.	-0.5871	0.8134
C	There's more less big fire at the Conference Center.	-0.6241	0.6616
	The explosion is somewhat terrifying.	-0.7753	0.9922
D	An extremely terrible fire burns the building.	-0.9285	0.8335
	I heard great nasty explosions.	-1	1

11.4.4 Valence and Arousal Thermometers

Two emotion thermometers, the valence T_v and the arousal T_a, can be used to observe the intensity of the user emotional state in a dialogue. These thermometers are plotted in the interval $[-1..1]$. Therefore, $v'_c = 2v_c - 1, v'_c \in [-1..1]$ and $a'_c = 2a_c - 1, a'_c \in [-1..1]$, where positive value means positive orientation and intensity and negative value means negative orientation and intensity. For every new user input, all thermometers are calculated using the following equations:

$$T_v(t+1) = \begin{cases} 1 & T_v(t) + v'_c \geq 1 \\ T_v(t) + v'_c + \varphi & \\ -1 & T_v(t) + v'_c \leq -1 \end{cases}$$

$$T_a(t+1) = \begin{cases} 1 & T_a(t) + a'_c \geq 1 \\ T_a(t) + a'_c + \varphi & \\ -1 & T_a(t) + a'_c \leq -1 \end{cases}$$

These values are considered as the user's current valence and arousal scores. The aggressive sign is defined as Boolean TRUE or FALSE of whether the parser finds any aggressive keyword in C. The constant value φ is the neutral factor of the user's emotional state, which works in two conditions: (1) if aggressiveness = FALSE and $v'_c = a'_c = 0$, the φ is positive and (2) if aggressiveness = TRUE and $v'_c = a'_c = 0$, the φ becomes negative.

11.4.5 System Design

Figure 11.8 show the class diagram of the emotionanalyzer module. Inside this module, the EmotionAnalyzer class analyzes the emotional loading of a set of concepts or text. For the text input, this class executes an additional process to chunk the input using the tagger module and knowledge from the wordnet module. The next is the EmotionAnalyzer class extracts both types of input into a set of Sentence objects, which consist of one or more Phrase objects. The Phrase class consists of one or more Word objects.

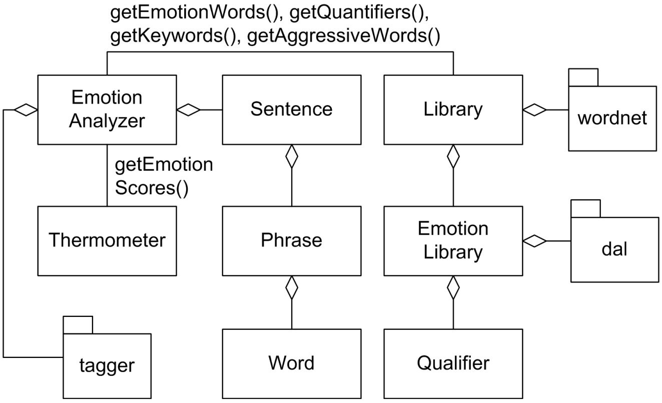


Figure 11.8: *The class diagram of the emotion analyzer*

To analyze the emotion loading of the input, the EmotionAnalyzer class uses the emotionlibrary module. The results of this analysis are the input valence and arousal scores. The EmotionLibrary class has a list of aggressive words, a list of (crisis) important keywords, a list of Quantifier objects, and a list of emotion words. The emotion words and their scores are stored in the dal module.

The Thermometer class receives the analysis results and uses them to update its value. The output of the EmotionAnalyzer class is the values of the instances of the Thermometer class. Depending on the source of the input, then the output is considered as the source's emotional state.

Chapter 12
Computed Situation Awareness

In which methods for the interpretation of a user message and the integration of multiple messages are described.

Context awareness involves the abstraction and understanding of context. In communication, context is perceived from shared information concerning some particular facts, subjects or events. In a significant sense, sensory input becomes information as the basis for context awareness. The perceived sensory stimulus can be examined and matched to a specific context. The stimulus includes what one sees, hears and experiences. It may be effected by one's beliefs, opinions and emotions. The result of such awareness is an action that is triggered based on known contexts. In HCI domain, situation awareness is dealing with the awareness of the logically interrelated contexts of a user. The current user context is interpreted as the current situation of the user. If a HCI system is able to perceive the situation of the user in his/her environment, consequently, it is able to adapt its behaviors (such as services, presented data and user interface) to that situation.

12.1 Situation Awareness

According to Endsley (1995), achieving situation awareness is a three stages process: perception, comprehension, and projection. Perception deals with monitoring, detection and recognition the status, attributes and dynamics of relevant concepts in the environment. Comprehension involves input recognition, interpretation and evaluation. It requires integrating the results of perception to understand how they will impact upon the individual's goals and objectives. This includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual. Projection implies the ability to project the future actions of the elements in the environment. It is achieved through the knowledge and then extrapolating this information to determine how it will affect future states of the operational environment.

In this chapter, the acquisition and interpretation of relevant concepts in the environment means decoding user communicated messages in order to have meaning. If a user message is regarded as a language, there is a need to know how the users communicate using this language. No matter what kind of language, here, an input is viewed as a spatial arrangement of concepts and their relations. The interaction between concepts (that are represented either by words or visual symbols) enriches progressively the semantic of the input. The meaningful structure is constructed as the assimilation of concepts and their relations into existing cognitive structures.

Figure 12.1 shows the application of the model framework of situation awareness into the proposed HHI framework. The *perceptor* builds concepts from a preprocessed input modality. It offers the advantages of coping with uncertainties at the input, as well as the ability to automatically mapping with the defined ontology. The language-to-concept parser is one of the perceptors that parses text into a set of concepts. Another perceptor developed in this research is the visual language-based interface. It has a parser that maps visual language-based inputs into concepts.

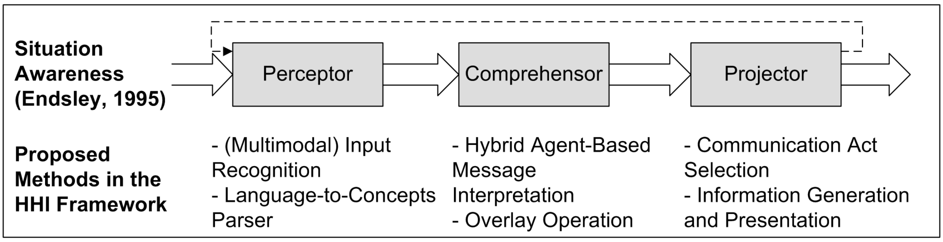

Situation Awareness (Endsley, 1995)
Perceptor
Comprehensor
Projector
Proposed Methods in the HHI Framework
- (Multimodal) Input Recognition
- Language-to-Concepts Parser
- Hybrid Agent-Based Message Interpretation
- Overlay Operation
- Communication Act Selection
- Information Generation and Presentation

Figure 12.1: *Applying the model framework of situation awareness into the HHI framework*

In sequence with these parsers, as a *comprehensor*, an agent-based message interpreter, an emergent self-organizing mechanism was designed to find coherent structures from activated world model concepts and parser inputs. It was designed based on the general architecture for simulating emergent human sensory (auditory) perception of Dor (2005). The simulator was developed based on the Copycat analogy-making model (Mitchell, 1993; Hofstadter, 1996). The model consists of cooperative and competitive interactions of auditory structure-building agents driven by plausible psychoacoustic grouping pressures. Incoming cues are treated as empty pieces of data to be interpreted only in conjunction with the emergence of higher-level structures.

Finally, the *projector* may use the concept activation to influence the building of structures among concepts on the user input space and the world model, trigger the user for additional inputs, supply context for the parser, and finally, when enough structure coherence produce a representation of the current interpretation of the fused modalities. Modules that deal with the selection of communication acts (see Section 13.2) and the output generation and presentation (see Section 13.3), are the projectors developed in this research.

12.2 Language-to-Concept Parser

In the task of automatically parsing text into concepts, the language-to-concept parser uses the following algorithm:

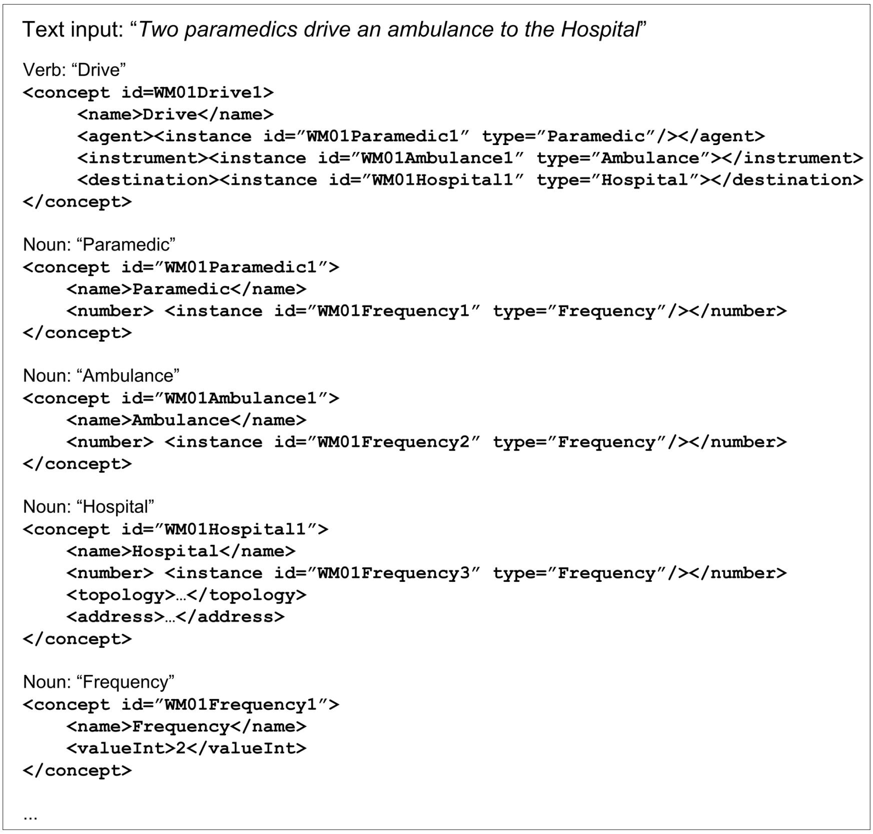
1. Chunk the text input into sentences and decompose the syntax structure of the sentences based on its thematic roles. To get the syntax structure of a sentence,

a tagger can be used. VerbNet, a verb lexicon with syntactic and semantic information for English verbs (Schuler, 2005), then, can be utilized for assigning the thematic roles into the syntax structure.

2. Replace all co-reference words (such as it, she, this) with the actual objects, all countable nominal values with numbers and all uncountable nominal values with available quantifiers.

3. Collect all words, search corresponding concepts in the ontology and instantiate these collected concepts.

4. Apply the relation between words based on their roles in the sentence to fill in the properties of their corresponding concepts. This process can be done since the properties of the classes in the ontology were also designed using thematic roles (see Chapter 9).



Text input: “*Two paramedics drive an ambulance to the Hospital*”

```
Verb: “Drive”
<concept id=WM01Drive1>
     <name>Drive</name>
     <agent><instance id=”WM01Paramedic1” type=”Paramedic”/></agent>
     <instrument><instance id=”WM01Ambulance1” type=”Ambulance”></instrument>
     <destination><instance id=”WM01Hospital1” type=”Hospital”></destination>
</concept>

Noun: “Paramedic”
<concept id=”WM01Paramedic1”>
    <name>Paramedic</name>
    <number> <instance id=”WM01Frequency1” type=”Frequency”/></number>
</concept>

Noun: “Ambulance”
<concept id=”WM01Ambulance1”>
    <name>Ambulance</name>
    <number> <instance id=”WM01Frequency2” type=”Frequency”/></number>
</concept>

Noun: “Hospital”
<concept id=”WM01Hospital1”>
    <name>Hospital</name>
    <number> <instance id=”WM01Frequency3” type=”Frequency”/></number>
    <topology>…</topology>
    <address>…</address>
</concept>

Noun: “Frequency”
<concept id=”WM01Frequency1”>
    <name>Frequency</name>
    <valueInt>2</valueInt>
</concept>

…
```

Figure 12.2: *An example of text input processing into instances of concepts*

All resulting instances found in the text input are sent to the message interpreter. The example in Figure 12.2 shows that the `Drive` concept has the `agent` property. This property can be filled with any concept that is a subclass of the `Actor` class. The `Paramedic` concept is a subclass of the `Actor` concept. The relations between these collected concepts build up the meaning that conveys the input.

12.3 Agent-based Message Interpreter

Each concept in the input is viewed as a representation of a puzzle piece (Figure 12.3(a)). Each piece can have properties to be filled in by other concepts. This creates relations between them. One or more relations may create a new concept. By relating all or some important pieces, the actual picture (the observer's view) can be revealed. This view is used to define a script. The selection of the right script is the goal of the interpretation process. A hybrid agent-based architecture is proposed to approach this puzzle arrangement problem. The approach comprises the (user) workspace, the system's ontology and scripts and the autonomous work of agents (Figure 12.3(b)).

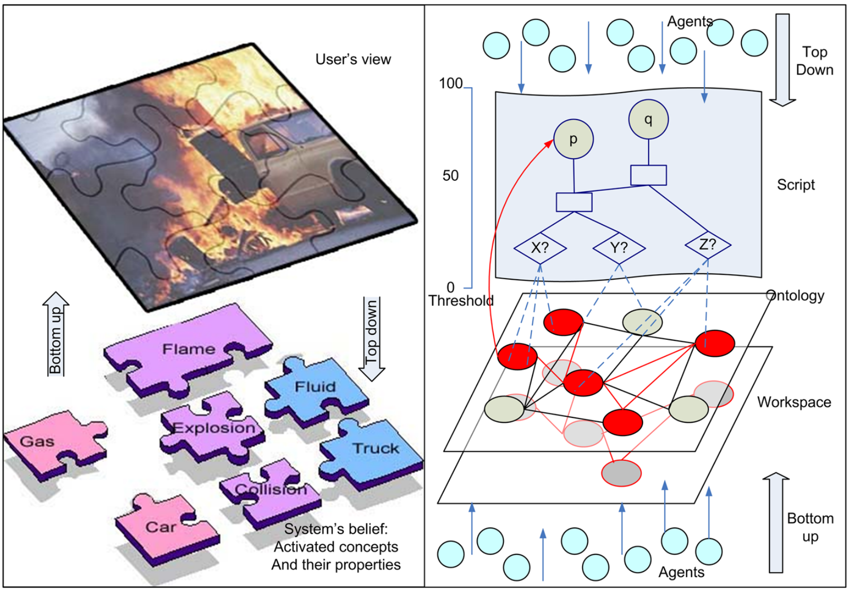
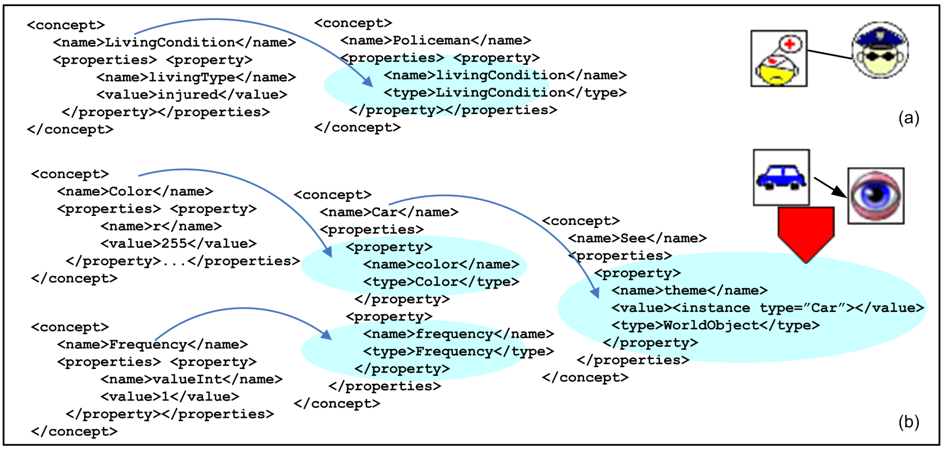


Figure 12.3: *(a) The concept-based puzzle arrangement metaphor and (b) the agent-based approach for message interpretation*

Workspace. The interpretation of an input is derived from purposive behavior emerging from interaction between multiple concepts in a user workspace. Workspace is where perceptual structures are built on top of the user's input. It contains all instances of concepts retrieved from the user's message. The model makes use of three types of structures: (1) description of concepts, (2) links representing the relation between concepts, and (3) groups of related concepts.



```
<concept>
   <name>LivingCondition</name>
   <properties> <property>
        <name>livingType</name>
        <value>injured</value>
    </property></properties>
</concept>
```

```
<concept>
   <name>Policeman</name>
   <properties> <property>
        <name>livingCondition</name>
        <type>LivingCondition</type>
    </property></properties>
</concept>
```

```
<concept>
   <name>Color</name>
   <properties> <property>
        <name>r</name>
        <value>255</value>
    </property>...</properties>
</concept>
```

```
<concept>
   <name>Frequency</name>
   <properties> <property>
        <name>valueInt</name>
        <value>1</value>
    </property></properties>
</concept>
```

```
<concept>
   <name>Car</name>
   <properties>
      <property>
         <name>color</name>
         <type>Color</type>
       </property>
      <property>
         <name>frequency</name>
         <type>Frequency</type>
       </property>
    </properties>
</concept>
```

```
<concept>
   <name>See</name>
   <properties>
      <property>
         <name>theme</name>
         <value><instance type="Car"></value>
         <type>WorldObject</type>
       </property>
    </properties>
</concept>
```

Figure 12.4: *(a) An example of two linked concepts and (b) an example of a group of linked concepts*

Agents. Agents are created to continually look for interesting properties and relationships among concepts and the message structures in the workspace using eq. 8.2 - 8.10 to define relationships of two concepts (as illustrated in Figure 8.5). On the basis of their findings, the agents build groups of related concepts, build relations between concepts, activate new concepts and fill in properties of the active concepts. These agents dynamically build (or break) all structures in the workspace. There are two types of agents, bottom-up and top-down. The bottom-up agents look for any structure they might find in the workspace. The top-down agents are sent by active scripts or concepts to look for or create instances of particular concepts. In the current implementation, three relationships are detected by a bottom-up agent:

1. A relation between two concepts. In Figure 12.4(a), the `Policeman` concept has a direct relationship with the `LivingCondition` concept.

2. A group of related concepts. In Figure 12.4(b), the relationship between the `See` concept and the `Car` concept occurs because the latter is a subclass of the `DynamicObject` concept. These concepts yields a message: "I see a red car".

3. An indirect relation between concepts. In Figure 12.4(b), the `See` concept has a direct relationship with the `Car` concept but it has an indirect relationship with the `Color` concept and the `Frequency` concept.

Ontology. The ontology holds predefined concepts in the form of a graph of concepts and properties. As agents find matching instances in the workspace, they activate their corresponding concept-property in the ontology. The activation of each concept-property affects the agent to consider it while working on a solution to a given problem. In this case is applying the law of proximity between concepts. When a concept in the ontology is activated, an agent will try to assign as many properties as possible by launching more agents to search for more values. As a result, an active concept will spread some activation to other relevant concepts. Furthermore, an active concept can activate one or more scripts. The activation levels of concepts decays at predefined intervals when agents fail to fill in their properties. By this mechanism, the system processes concepts that are dynamically deemed relevant, while still allowing new concepts to be perceived by constantly launching agents.

Scripts. An active concept in the ontology can activate one or more scripts. In any process, there can be many competing scripts. An active script can launch more agents to evaluate all concepts in the conditions of each frame. These agents check the instances of these concepts in the user workspace. They may propose to instantiate concepts in an active script that are not yet defined in the user workspace based on the relationship of these concepts with known concepts that have activated the corresponding script. As more concepts are active in the ontology, more key concepts represent most common to some competing scripts. As a result, certain scripts become impossible and are removed from competition. This process continues until, there is only one script left. At this point, it is assumed that a specific scenario is identified and the interpretation of the user's input is produced. If the last script is also impossible, the rest unevaluated scripts are evaluated. The process is repeated. Additionally, an active concept may activate an evaluated script and make it competitive again. This entire process will be stopped if there is not any script left to be evaluated. At this point, although a specific scenario is not identified If none of the scripts is selected, but the interpretation process has produced a list of linked concepts.

12.4 Multiple Message Integrator

As part of the work of a comprehensor, an overlay operation (Alexandersson & Becker, 2003) is utilized to form an aggregated world view based on interpretations of multiple messages. It is a formal operation based on unification of typed feature structures - TFS (Carpenter, 1992) (Figure 12.5(b)). Overlay can be seen as putting two typed feature structure - covering and background - on top of each other (Figure 12.5(a)). The background can be viewed as the old state of the world, while the covering is an incoming message. The steps to handle all inputs are:

1. ***Map concepts into*** TFS. This works as follows: (a) collect all active concepts, (b) fill all properties with actual values (from the instances), and (c) if a TFS is denoted as $\{t, [f_1 : v_1, \ldots, f_n, v_n]\}$ where t is the type, $[f_1 \ldots f_n] \in F$ are the type features and $[v_1 \ldots v_n] \in V = A \cup S$ are the values where S is the set of typed feature structures, map all concepts into t_i with $\{f_{i1} : v_{i1} \ldots f_{in} : v_{in}\}$ as the set of property-value pairs. Figure 12.6 shows an example of the mapping result.

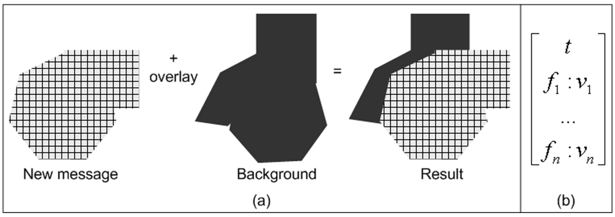

+
overlay
=
New message
Background
Result
(a)
$\begin{bmatrix} t \\ f_1 : v_1 \\ \ldots \\ f_n : v_n \end{bmatrix}$
(b)

Figure 12.5: *(a) The schematic view of overlay operation and (b) formal representation for denoting a TFS*

2. ***Assimilate the background to the covering message.*** Let a finite set of types $T = \{t_1, \ldots, t_n\}$ with $t_i \preceq t_j$ means t_i is a super-type of t_j. The assimilation of the covering $c = \langle t_c, \{f_1 : v_1, \ldots, f_n : v_n\}\rangle$ and background $b = \langle t_b, \{g_1 : w_1, \ldots, g_n : w_n\}\rangle$ works in two cases: (1) if $t_c \preceq t_b$, then $\alpha(c,b) := ((\langle t_b, \{f_1 : v_1, \ldots, f_n : v_n\}\rangle, b)$ and otherwise (2) with $g_i = lub(t_c, t_b)$, $\alpha(c,b) := (c, \langle t_c, \left\{g_i : w_i, \ldots, g_j : w_j\right\}\rangle)$. Two types can be incompatible if they are not related via a super-type relation. The operation also considers a predefined (threshold) distant reference for the time and (geographical) location of the two types (that is when and where their corresponding concepts are initialized and placed). The transformation may remove some incompatible features (at least for b), however, it does not affect the values of features. A function $lub(t_c, t_b)$ computes the most specific common super-type for two types of c and b.

15:01:07 An observer reports: "A truck is on fire on Provicialeweg", in XML form:

```
<fire>
  <causal>
    <location> ...</location>
    <beginTime>...</beginTime>
  </causal>
  <location>...</location>
  <beginTime>...</beginTime>
  ...
</fire>
...
```

Initial world model:

$$TFS : \begin{bmatrix} FIRE \\ causal : \begin{bmatrix} TRUCK \\ location : \begin{bmatrix} ADDRESS \\ streetName : \text{Pr } ovicialeweg \end{bmatrix} \\ beginTime : \end{bmatrix} \\ location : \ldots \\ beginTime : \ldots \end{bmatrix}$$

Figure 12.6: *Mapping concepts (in XML) into TFS*

3. ***Apply overlay operation.*** Here, the overlay operation is applied on the covering $c = \langle t_c, \{f_1 : v_1, \ldots, f_n : v_n\}\rangle$ and background $b = \langle t_b, \{g_1 : w_1, \ldots, g_n : w_n\}\rangle$:

$$overlay(c,b) = overlay'(\alpha_c(c,b), \alpha_b(c,b)) \tag{12.1}$$

   (a) If recursion:

$$overlay'(c,b) = \langle t_c, \left\{p_i : h_i | p_i = c_j = b_k, h_i = overlay\left(f_j, g_k\right)\right\}\rangle \tag{12.2}$$

where $f_j, g_k \in S$

(b) If the covering and the background have (atomic) values:

$$overlay'(c,b) = \langle t_c, \left\{ p_i : h_i | p_i = c_j = b_k, h_i = f_j \right\} \rangle \tag{12.3}$$

where $f_j, g_k \in A$

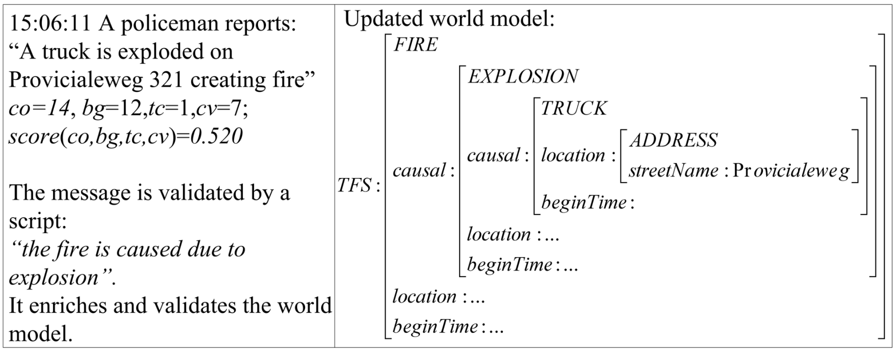
(c) If a feature is absent in the background:

$$overlay'(c,b) = \langle t_c, \left\{ p_i : h_i | p_i = c_j, h_i = f_j, p_i \neq b_k, 1 \leq k \leq m \right\} \rangle \tag{12.4}$$

(d) If a feature is absent or has no value in the covering:

$$overlay'(c,b) = \langle t_c, \left\{ p_i : h_i | p_i = b_k, h_i = g_k, p_i \neq c_j, 1 \leq j \leq n \right\} \rangle \tag{12.5}$$

If $c \neq b$ then $overlay(c,b) \neq overlay(b,c)$ and if c and b are `unifiable` then $overlay(c,b) = overlay(b,c) = unify(c,b)$.



15:06:11 A policeman reports: "A truck is exploded on Provicialeweg 321 creating fire" *co=14*, *bg*=12,*tc*=1,*cv*=7; *score*(*co,bg,tc,cv*)=*0.520* The message is validated by a script: *"the fire is caused due to explosion".* It enriches and validates the world model.	Updated world model: $TFS: \begin{bmatrix} FIRE \\ causal: \begin{bmatrix} EXPLOSION \\ causal: \begin{bmatrix} TRUCK \\ location: \begin{bmatrix} ADDRESS \\ streetName: \mathrm{Pr}ovicialeweg \end{bmatrix} \\ beginTime: \end{bmatrix} \\ location: ... \\ beginTime: ... \end{bmatrix} \\ location: ... \\ beginTime: ... \end{bmatrix}$

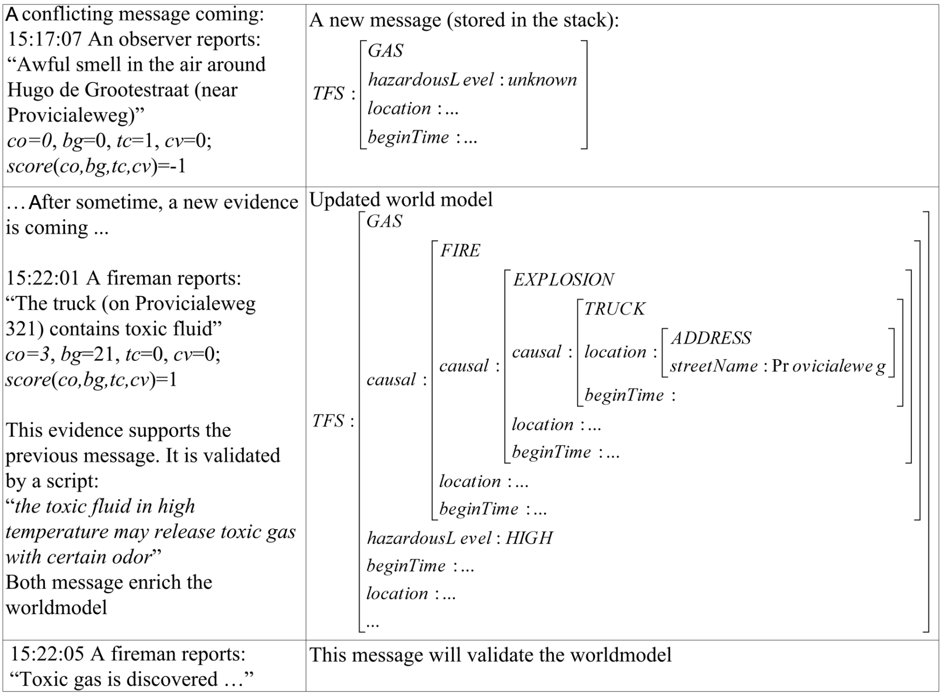
Figure 12.7: *An incoming message enriches and validates the world model*

4. ***Calculate overlay score.*** This score is defined to reflect how well the covering fits the background in terms of non-conflicting features (Pfleger et al., 2002):

$$score\left(co, bg, tc, cv\right) = \frac{co + bg - (tc + cv)}{co + bg + tc + cv} \in [-1 \ldots 1] \tag{12.6}$$

where co is the number of values stemming from the covering c for cases eq. 12.2, eq.12.3 and eq.12.4; bg is the number of values stemming from the background b for cases eq. 12.2, eq.12.3 and eq.12.5; tc is the number of not-identical types of both c and b that are identified using $\alpha(c,b)$; and cv is the number of conflicting values (case eq. 12.3 - when the value of a feature of b is overwritten). The $score = 1$ indicates the feature structure is unifiable and the $score = -1$ indicates all information from b has been overwritten. All scores between these two extremals indicate that c more or less fits b; the higher the score the better the fit.

5. ***Enrich or validate the state of the world.*** The background message is updated. The current implementation processes all messages that have the $score > 0$. Messages with $score \leq 0$ are stored in a stack until some evidences support them (validated by a script). In particular, an activation level of active concepts in the ontology is defined. The level will grow if new messages include these concepts and decay at predefined intervals, otherwise. By this mechanism, only up-to-date (observed) concepts are active in the world model. The world model is expected to be constructed based on reliable messages; while those unreliable messages by the mechanism eventually will be discarded. In the current implementation, the first input message is treated as the initial background.



A conflicting message coming: 15:17:07 An observer reports: “Awful smell in the air around Hugo de Grootestraat (near Provicialeweg)” *co=0*, *bg*=0, *tc*=1, *cv*=0; *score*(*co,bg,tc,cv*)=-1	A new message (stored in the stack): $TFS : \left[\begin{array}{l} GAS \\ hazardousLevel : unknown \\ location : ... \\ beginTime : ... \end{array}\right]$
…After sometime, a new evidence is coming ... 15:22:01 A fireman reports: “The truck (on Provicialeweg 321) contains toxic fluid” *co=3*, *bg*=21, *tc*=0, *cv*=0; *score*(*co,bg,tc,cv*)=1 This evidence supports the previous message. It is validated by a script: “*the toxic fluid in high temperature may release toxic gas with certain odor*” Both message enrich the worldmodel	Updated world model $TFS : \left[\begin{array}{l} GAS \\ causal : \left[\begin{array}{l} FIRE \\ causal : \left[\begin{array}{l} EXPLOSION \\ causal : \left[\begin{array}{l} TRUCK \\ location : \left[\begin{array}{l} ADDRESS \\ streetName : Provicialeweg \end{array}\right] \\ beginTime : \end{array}\right] \\ location : ... \\ beginTime : ... \end{array}\right] \\ location : ... \\ beginTime : ... \end{array}\right] \\ hazardousLevel : HIGH \\ beginTime : ... \\ location : ... \\ ... \end{array}\right]$
15:22:05 A fireman reports: “Toxic gas is discovered …”	This message will validate the worldmodel

Figure 12.8: *Storing a conflict message in a stack until its supported evidences come*

Figure 12.6 shows a set of concepts about a fire accident with a truck at a certain address. In Figure 12.7, the message is validated by a policeman, who is already inspecting the scene. He adds extra information about the cause of the fire, which is supported by a script. The report about the address of the event has enriched the world model. Figure 12.8 depicts the internal representation when a new message is coming from a citizen about the smell around his area. This message does not have any connection with the formulated world model. It is then stored in the stack until other evidences can support it. A new evidence is coming from a fireman and evaluated by a script. In this case, the toxic fluid stored in the truck may release toxic gas due to the explosion and fire. Both messages, then, can enrich the world model.

The interpretation process and results do not include any private information of users, except their geo-reference location. All observers are treated as (anonymous) actors doing a certain action in a certain location. The information about these actors, combining with information about specific objects and events appeared in the world, builds up a scenario of a certain crisis event, which is validated by the scripts.

12.5 System Design

Figure 12.9 shows the class diagram of the `fusion` module. The `Analyst` class receives a list of active `Report` objects from the `ReportProcessor` class. The input can be an arrangement of visual symbols or text. Using the `ontology` module and the `iconlibrary` module, the `Workspace` class collects the corresponding concepts of the visual symbols. The text is extracted by the `LanguageToConceptParser` class into concepts using the `tagger` module, the `verbnet` module and the `wordnet` module. The results are added into the previous concepts built by the `Workspace` class.

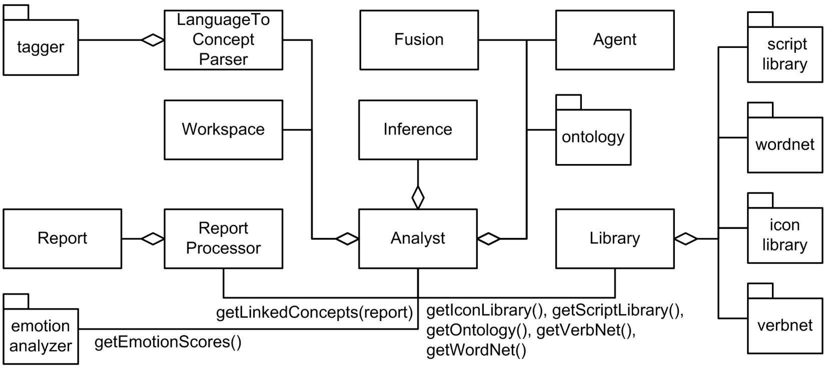


Figure 12.9: *The class diagram of the fusion module*

The `Analyst` class has `Agent` objects that apply all possible relations between concepts found in the `Workspace` class based on classes in the `ontology` module and the law of proximity (see Section 8.4). The `Agent` objects also apply such relations to concepts from the `Inference` class. Using the `script` module, this `Inference` class selects the most appropriate script for the input concepts. The result of these agents' work is a set of linked concepts. It is treated as a new incoming message by the `Fusion` class. This class integrates the new message and updates the system's belief that was built based on previous messages.

Chapter 13

Multimodal Information Generation and Presentation

In which research in developing interaction manager and output fission modules is described. Appropriate multimodal and context-sensitive information is presented tailored to the user by the employment of ontology from and to multiple users, multiple devices and multiple modalities.

What we do as we converse is constrained by what is done by our interlocutors. We tailor what we say specifically for our conversational partners. We adapt to our partners over the course of a conversation. Communication always occurs, not in a isolation, but in context of the utterances of the other people. As a feedback, their contributions may constrain our response. This includes both content and behavior (the way we connect to each other). Although the course of the conversation may be unpredictable, misunderstandings are often handled by giving feedback to each other. In this way, better comprehension and effective communication can be achieved. Moreover, relationship between people can be improved. Similar mechanisms are expected taking place in HCI.

13.1 Intelligent Information Presentation

Intelligent information presentation relates to the ability of a system to automatically produce interactive information presentation (Stock & Zancanaro, 2005). It offers a user-system collaborative interaction for the retrieval of relevant information. This takes into account knowledge about the user and the context in which the interaction takes place. The process of elaborating a presentation starts with selecting a communication act based on communication plans and intentions, then specifying the content based on some internal knowledge or experience of previous presentation, designing the presentation and finally allocating coordinated information across available means of communication. This entire process is dealing with conveying the information in the most appropriate way. Knowledge about user, system beliefs, dialogue history and communication device constraints are considered in the process.

Deciding which communication act to pursue as well as how and when is usually assigned to an interaction manager (IM or also called a dialogue action manager in a dialogue system). An IM of an adaptive system works by building a model of the user's goals, the results retrieved, the state of the dialogue and the system's response at each turn of the dialogue - and using these throughout the interaction for adaptation of the output. By keeping a model of the user's characteristics, the output can be adapted to aid the user in navigating and filtering information that best suits the intended goals.

To select a specific act in an interaction flow, one of the approaches is by using a set of interaction strategies. These strategies can be formed in tables (Zue et al., 2000), frames (Roberta Catizone & Wilks, 2003) or task hierarchies (Potamianos et al., 2000). Basically, the approach uses a set of rules and a state machine to analyze the structure of an interaction and to select a specific strategy McTear (2002). Others use statistical techniques. Keizer (2003) have used Bayesian Networks to model the interaction, while (Bosma & Andre, 2004) have included emotions into the network to disambiguate communication acts. The POMDP technique has been used by Bui et al. (2007) to model the user's emotional state, intentions and hidden states in a computation to select the next action. Vark et al. (1996) have analyzed over 5000 recorded dialogues of public transport information and classified them into a set of communication acts. The predefined acts were interconnected by directed links. To such a link, a probabilistic value was assigned derived from the corpora analysis (Rothkrantz et al., 2000). An optimal communication strategy can be defined as selecting the system's action with the highest probability over the course of an interaction.

The research in embedded character agents has addressed the impact of emotion in the interaction strategy and service performance (Cassell, 2000). Here, two types of emotion expressions were considered: (1) to communicate internal states and intentions to others and (2) to contribute to the activation and regulation of emotion experiences (Pelachaud & Bilvi, 2003). Research in this field usually concerns the nature of emotion felt, such as intensity, valence and intention. Others take into account interaction of contextual factors, such as motive, personality, personal characteristics, role relationship and appraisal for others.

A presentation planner retrieves the unified information and specifies to enable for two types of generations. The first is Natural Language Generation. This branch of NLP deals with the automatic production of texts. The language generation typically uses some linguistics approaches, such as TAG (Joshi & Vijay-Shanker, 1999) in SmartKom, n-grams and CCG (Steedman & Baldridge, 2005) in COMIC, and template-based in Match and AdApt. The template-based language generation uses a limited relational memory, in which concepts are represented in list structures of keyword or word categories and simple relations of pattern operations rules that exist within these structures. Eliza (Weizenbaum, 1966) is probably the oldest dialogue system based on this approach. In 1995, A.L.I.C.E improved the Eliza system in focusing on a certain topic and generating responses based on previous conversation using AIML (Wallace, 2003). AIML (Artificial Intelligent MarkUp Language) is an extended-XML script that provides specifications for pattern input matching and reply generation.

The second type is display generation, which deals with graphic productions. Both language and graphic productions usually start with investigating communication acts, then the dynamic choice of what to say, the planning of the overall rhetorical structure of the output and finally the actual realization of output across media. For coordinating virtual agents' behaviors, graphical displays and speech, techniques like stack-based in Match, schema-based in COMIC and XIMS, frame-based in AdApt and rule-based in MACK, are applied (see Section 3.3.1).

Multimedia services that are generated in a coordinated fashion on a combination of multiple modality devices, can provide users with more choices to use different modalities and devices and to find their own optimal experiences (Maybury & Wahlster, 1998). The integration interaction means available on the user side into the HHI provides challenges on several levels, such as: (1) how to accommodate the flexible switching of communication modes and devices, which can have different capabilities of the physical device and of the provided I/O modalities, (2) how to adapt the application and information presented to user context variables (such as profile, emotion, location) which may change over time due to mobility and dynamic environment, and (3) how to offer a unified view of services under such technology constraints and dynamic changes. These challenges yield serious interface and interaction design issues.

13.2 Interaction Management

The knowledge engineering in the field of crisis management has resulted in seven interaction frames in Table 13.1 (see detailed in Section 9.2). The analysis was based on the behavior of a human crisis hotline dispatcher. These frames have been used to develop the interaction strategies of the research demonstrator. Its discourse model is developed similar to the approach of Rothkrantz et al. (2000). Each frame consists of one or more slots. The goal of the user system interaction is to fill the frames and their slots in an efficient manner. This interaction is handled by an IM module.

Table 13.1: *Example interaction frames and their slots in the field of crisis management*

Frame	Slots
user	{name, address(street, no, city, postcode), phone-no, role["VICTIM", "WITNESS", "PARAMEDIC", "FIREMAN", ...], emotion-type}
problem	{type["FIRE","TRAP",’SICK", ...], status, desc}
time	{hour, minute, second, day, month, year}
location	{street, no, city, postcode}
involved_party	{type["VICTIM", "SUSPECT", "PARAMEDIC", ...], status, address(street, no, city, postcode), name, phone-no}
reason	{type["HIT", "EXPLOSION","GAS-LEAKING", ...]}
weapon	{type["GUN", "KNIFE","BOMB", ...]}
urgency	{type[HIGH, MEDIUM, LOW]}

The IM module works based on predefined interaction strategies. It is constructed using goal-directed principles. The selection of an appropriate communication act is based on a comparison of a representation of the goal state and the current state. The comparison is based on the evaluation of how much interaction steps will reduce the distance to the goal state. The input analyzer and the communication act selector in the IM module (Figure 13.1) are explained in the following subsections.

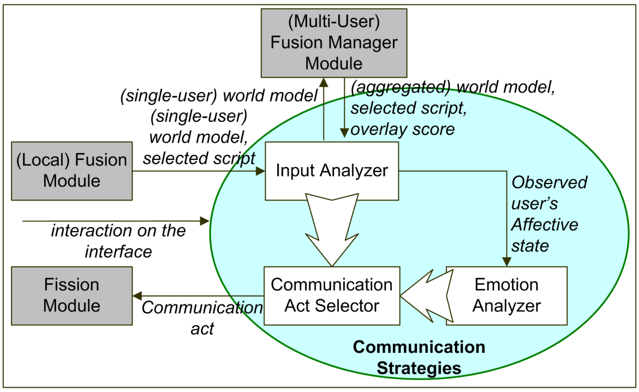

(Multi-User) Fusion Manager Module
(single-user) world model
(aggregated) world model, selected script, overlay score
(single-user) world model, selected script
(Local) Fusion Module
Input Analyzer
Observed user's Affective state
interaction on the interface
Fission Module
Communication act
Communication Act Selector
Emotion Analyzer
Communication Strategies

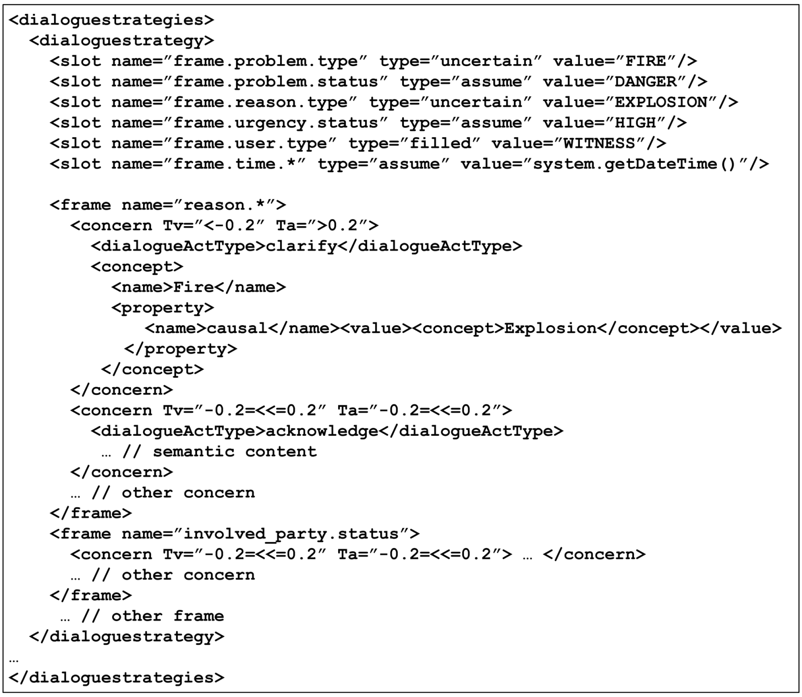
Figure 13.1: *The architecture of the interaction manager module*

13.2.1 Input Analyzer

Figure 13.2 shows an example of the strategies in the field of crisis management. The value of the `<frame>` tag marks the current communication action and the value of `<concern>` tag marks the user's emotional state. Each strategy provides some possible communication acts that can be selected depending on the value of both tags. In the current implementation, only the communication act `scenario` is triggered directly by the user interaction on the interface.

All frames are empty at the beginning of an interaction. Every state represents the known knowledge at that time. A new state occurs after the world model is updated based on the user message. The input analyzer receives an updated world model from a fusion module. In a multi-user system, this component sends the world model of a specific user to a multi-user-message integration module (via a predefined network). The component, then, will receive an integrated multi-user world model. If the integrated world model has a different conclusion of the current situation from the user's world model, the input analyzer can trigger the emotion analyzer component to check the user's emotional state.

Based on the new input, the input analyzer fills the value of slots of every known frame. It decides the type of the value whether it is extracted from the input (`filled`), deducted from the input interpretation (based on known concept-property relations - `assume`) or uncertain (`uncertain`). For example, if the system receives an input `"A truck is on fire"`, this component will set `frame.problem.type="FIRE"`



```
<dialoguestrategies>
  <dialoguestrategy>
    <slot name=”frame.problem.type” type=”uncertain” value=”FIRE”/>
    <slot name=”frame.problem.status” type=”assume” value=”DANGER”/>
    <slot name=”frame.reason.type” type=”uncertain” value=”EXPLOSION”/>
    <slot name=”frame.urgency.status” type=”assume” value=”HIGH”/>
    <slot name=”frame.user.type” type=”filled” value=”WITNESS”/>
    <slot name=”frame.time.*” type=”assume” value=”system.getDateTime()”/>

    <frame name=”reason.*”>
      <concern Tv=”<-0.2” Ta=”>0.2”>
        <dialogueActType>clarify</dialogueActType>
        <concept>
          <name>Fire</name>
          <property>
             <name>causal</name><value><concept>Explosion</concept></value>
           </property>
         </concept>
      </concern>
      <concern Tv=”-0.2=<<=0.2” Ta=”-0.2=<<=0.2”>
        <dialogueActType>acknowledge</dialogueActType>
         … // semantic content
      </concern>
      … // other concern
    </frame>
    <frame name=”involved_party.status”>
      <concern Tv=”-0.2=<<=0.2” Ta=”-0.2=<<=0.2”> … </concern>
      … // other concern
    </frame>
     … // other frame
  </dialoguestrategy>
…
</dialoguestrategies>
```

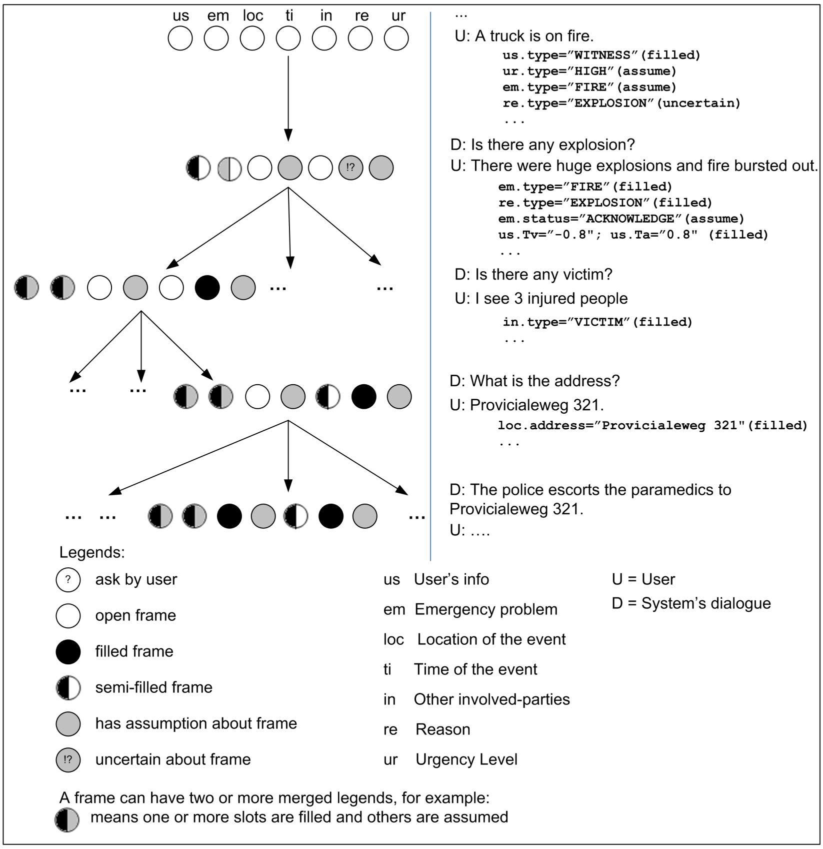
Figure 13.2: *A communication strategy (in XML format) in the field of crisis management*

and this slot's type is "filled". Figure 13.3 shows that the problem frame becomes semi-filled in the first interaction.

13.2.2 Communication Act Selector

In every state of a user-system interaction, there are different ways to continue an interaction. One way is based on the user's current emotional state, which is indicated by the `<concern>` tag. T_v is the value of the valence thermometer and T_a is the value of the arousal thermometer (see Section 11.4 for their calculation). The IM module retrieves both values from the user model (see Section 9.3.2).

Another way to select the next action is based on an evaluation of whether by selecting an action, a maximum slot-filling is achieved. For this purpose, firstly, optional frames and slots were indicated based on the nature of the emergency problem. For example, in the case of the `problem.type = FIRE` some optional slots, such as `weapon.type`, may not be filled in. Secondly, heuristics rules have been designed, such as: (a) if there is a (non-empty) subset of open frames and a (non-empty) subset of filled frames, ask for information covering as much as the open frames as possible; (b) if one frame in the current state is not completely filled, immediately ask for the missing information to solve the ambiguity concerning this frame; (c) if the subset of not complete filled frames contains more than one slot, handle the individual slots one after the other; and (d) as long as new information can be provided, assumptions of the values are not verified.



...
U: A truck is on fire.

```
us.type="WITNESS"(filled)
ur.type="HIGH"(assume)
em.type="FIRE"(assume)
re.type="EXPLOSION"(uncertain)
...
```

D: Is there any explosion?
U: There were huge explosions and fire bursted out.

```
em.type="FIRE"(filled)
re.type="EXPLOSION"(filled)
em.status="ACKNOWLEDGE"(assume)
us.Tv="-0.8"; us.Ta="0.8" (filled)
...
```

D: Is there any victim?
U: I see 3 injured people

```
in.type="VICTIM"(filled)
...
```

D: What is the address?
U: Provicialeweg 321.

```
loc.address="Provicialeweg 321"(filled)
...
```

D: The police escorts the paramedics to Provicialeweg 321.
U: ....

Figure 13.3: *Communication model from the system perspective (circles represents frames)*

Figure 13.3 shows the differences in the known knowledge in every communication state. Given a set of possible (open) frames and slots to choose, the selector selects associated frame names and slot names from the sets of component strategies. The most specific set of associated frame and slot names are matched first before any other frames or slots or default. Within a selected `frame` tag, given the current user emotional state, this component selects and returns the most appropriate response based on the value of the `concern` tag. For example, in Figure 13.2, if the user current emotion is considered negative (in this case $T_v < -0.2$) and the selected frame is `reason`, the selected communication act is `clarify`.

13.2.3 System Design

Figure 13.4 shows the class diagram of the interactionsmanager module. In this module, the InteractionManager class receives a list of linked concepts from the fusion module. The fusionmanager module is a subclass of the fusion module, which handles the integration of multiple messages from multiple users. Both modules are described in Chapter 12.

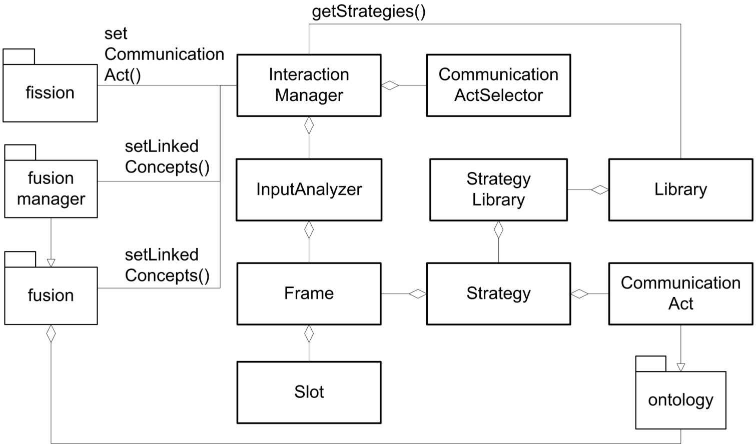


Figure 13.4: *The classdiagram of the interaction manager*

Based on the input linked concepts, The InteractionManager class uses the InputAnalyzer class to fill the Slot objects of every available Frame object. The CommunicationActSelector class uses this knowledge to match the current list of Frame objects for selecting the most appropriate CommunicationAct object. In this process, it uses the strategylibrary module from the Library class. The selected communication act is sent to the fission module, which will use it for generating and presenting information to the user. This process is described in the next section.

Using an XML notation of communication strategies, the StrategyLibrary class in the strategylibrary module generates a set of Strategy objects. For a set of possible known Frame and Slot objects, a Strategy object can be defined by some CommunicationAct objects with their corresponding Concept objects that may be available in the input linked concepts. The classes in the ontology module define the CommunicationAct classes.

13.3 Output Generation and Presentation

The literal meaning of *multimodal fission* refers to production by spontaneous division of the modalities (such as sound and video image) into two or more parts (such as speech, intonation, lip movements, facial expressions and the like) each of which grows into a complete correlated multimodal presentation of a meaning. The multimodal information generation and presentation in this research combines language, visual-language and graphics. It receives and investigates the communication act for a specific user from an IM module. Both the content information and its presentation are dedicated for a certain user and the knowledge (beliefs) constructed so far. The presentation of the output can be triggered both by user interaction or by a change in the world model. The different components in the fission module (Figure 13.5) are explained in the following subsections.

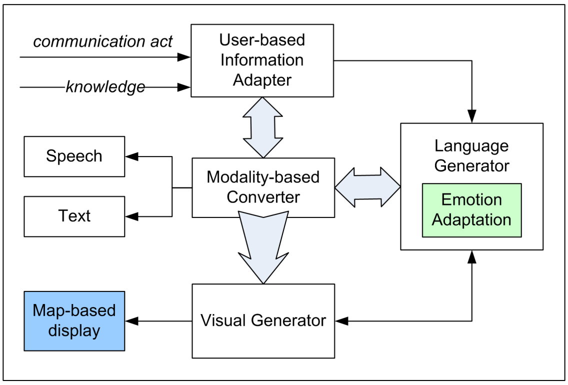


Figure 13.5: *The architecture of the fission module*

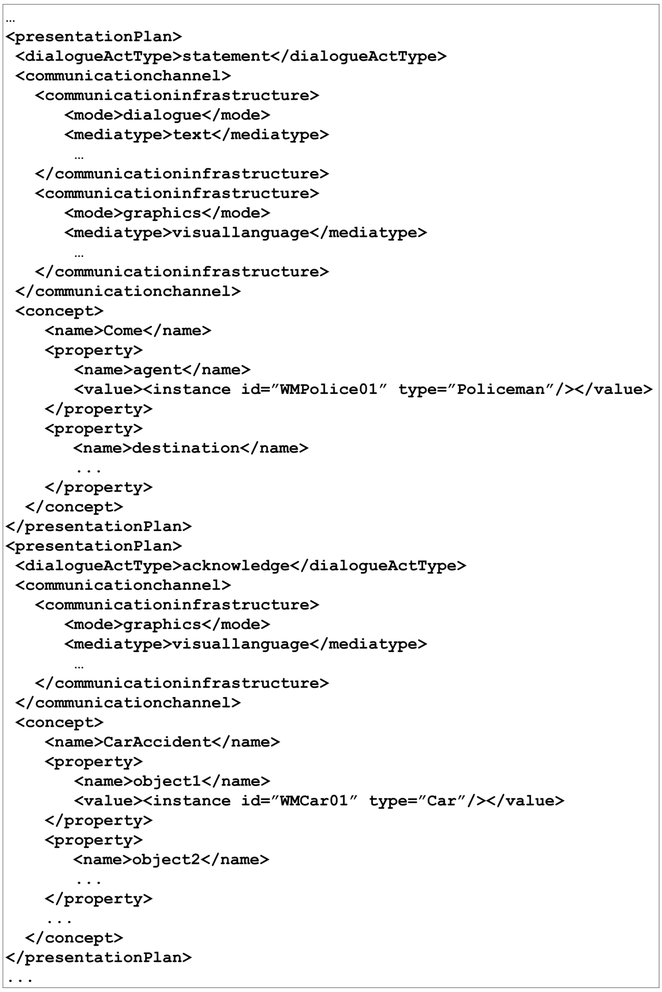
13.3.1 User-based Information Adapter

With direct access to three knowledge sources: the user knowledge (who, where, what the emotional state of and what the communication device of the user), the task knowledge (communication acts - general ideas of how tasks should be done) and the world knowledge (world facts and events), a user adapter decomposes the communication acts into presentation plans. This decomposition process includes contextual information (such as user model, task model and world model) and technical information (such as user preferences, output end-point constraints and availability of modalities). Here, the availability of real time context and technical information from a global representation is taken as an assumption while this component is actually planning the presentation tasks. The input representation includes the characteristics of modality services of the user's communication devices based on the device's capabilities and characteristics. Using this knowledge, the component plans the presentation contents based on the following contextual information of the user.

User role: dealing with filtering the right information contents for the right user. For example, predefined flags on properties of concepts in the ontology can be specified, such as public (available for all including general publics), protected (available for rescue workers and crisis room operators), and private (available for a certain rescue worker team or only crisis room operators).

User location: dealing with locating a new perspective area of the world knowledge based on the current position. The perspective view of a user may differ from the perspective area of the world model. Some information may become not important because it is far away from the user.

Interaction content: dealing with selecting the most up-to-date information to be presented to the user using this user's interaction history as a reference.



```
…
<presentationPlan>
 <dialogueActType>statement</dialogueActType>
 <communicationchannel>
   <communicationinfrastructure>
      <mode>dialogue</mode>
      <mediatype>text</mediatype>
       …
   </communicationinfrastructure>
   <communicationinfrastructure>
      <mode>graphics</mode>
      <mediatype>visuallanguage</mediatype>
       …
   </communicationinfrastructure>
 </communicationchannel>
 <concept>
    <name>Come</name>
    <property>
       <name>agent</name>
       <value><instance id=”WMPolice01” type=”Policeman”/></value>
    </property>
    <property>
       <name>destination</name>
       ...
    </property>
  </concept>
</presentationPlan>
<presentationPlan>
 <dialogueActType>acknowledge</dialogueActType>
 <communicationchannel>
   <communicationinfrastructure>
      <mode>graphics</mode>
      <mediatype>visuallanguage</mediatype>
       …
   </communicationinfrastructure>
 </communicationchannel>
 <concept>
    <name>CarAccident</name>
    <property>
       <name>object1</name>
       <value><instance id=”WMCar01” type=”Car”/></value>
    </property>
    <property>
       <name>object2</name>
       ...
    </property>
    ...
  </concept>
</presentationPlan>
...
```

Figure 13.6: *An example of a set of presentation plans (in XML format)*

User emotion: dealing with investigating the user's current emotional state. The component passes this information to the language generation component.

Available modalities: dealing with selecting the output modality to convey the output messages. Knowing characteristics of modality services of user communication devices, this component uses a set of negotiation policies to negotiate and selects output modalities based on the available output modalities and the input modalities used by the user. The current research only concerns text (and synthesized speech) and visual language-based output. However, other policies can be added, for example, by using reasoning to suggest the best set of modalities for a certain concept. The reasoning may also consider other contexts, such as the user's profile, the user's current task, the ambient of the user's location and the user's current emotional state.

Figure 13.6 shows a part of presentation plans resulted by this component. In this example, the first plan is to display the concepts from the communication act using both language presentation and visual presentation. The second plan is to display the concepts from the world model on the visual language-based interface.

13.3.2 Modality-based Converter

A modality-based converter uses the results of the user adapter to design the presentation and to allocate information across the production of language and visual display. The IM module may include multiple concepts within a selected communication act. The converter divides these concepts to be a set of sequenced segments. All segments will be linked and synchronized with concepts that are processed in both production components. Other active concepts in the world knowledge that are necessary to be displayed but are not linked to any language generation segment are passed to the visualization generation. These concepts will be displayed directly on the user's interface. This converter component also informs the visual generator about those concepts that are necessary to be updated since they are not longer active.

A queue of presentation segments is processed by the language generator. The modality-based converter then estimates the time needed for each processed segment. This depends on the chosen output modality. The component controls the output using this timing information to create a full schedule for the turn of each segment to be generated. Each language segment will be generated one by one synchronized with the display (of the active or the highlighted concepts) on the interface.

The schematic view in Figure 13.7 shows the internal representation of the user adapter and the modality-based converter. In this example, the concept 4 is hidden because its location is out of the new user's perspective area. The concept 1 and 2 will be removed from the interface since their corresponding concepts in the updated knowledge do not exist anymore. Although the corresponding concept of the concept 5 (on the user's interface) in the updated knowledge is not available anymore, the component does not remove this concept since it is inputted by the user. The concept 6 will be displayed on the interface without any language output. The length of time unit of each presentation plan depends on the schedule of each output generation in the presentation plan.

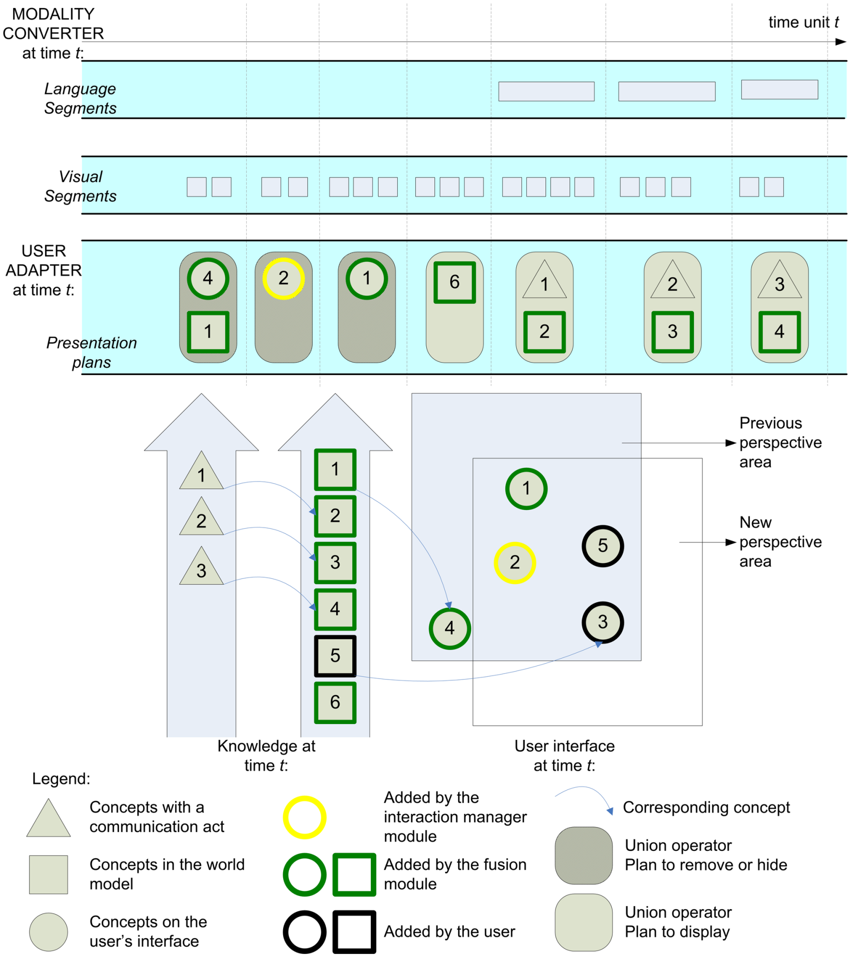
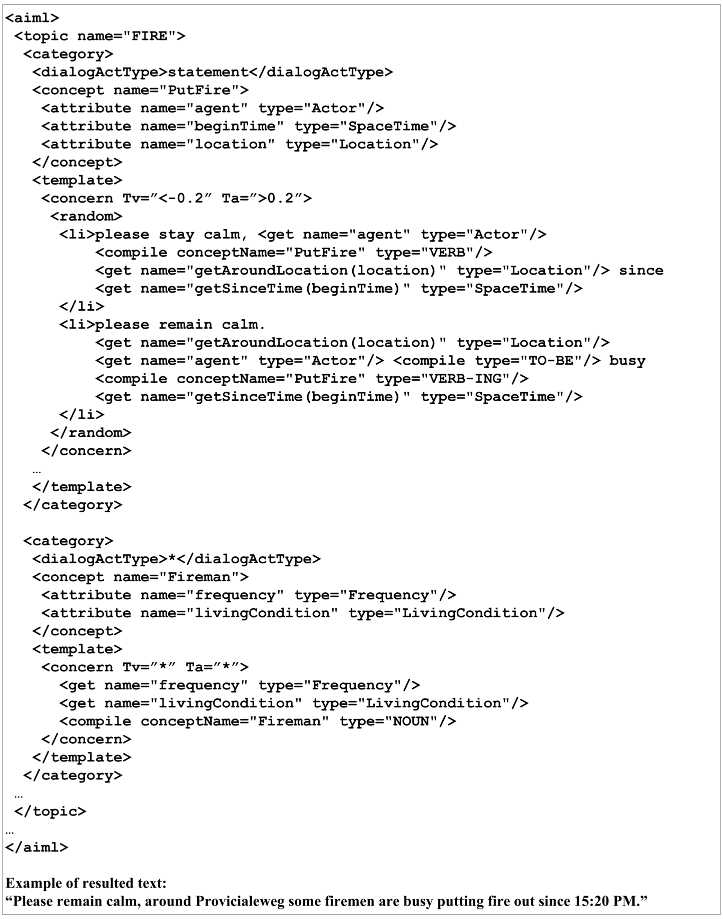

MODALITY CONVERTER at time t:
time unit t
Language Segments
Visual Segments
USER ADAPTER at time t:
Presentation plans
4
1
2
1
6
1
2
2
3
3
4
1
2
3
1
2
3
4
5
6
Previous perspective area
New perspective area
1
2
5
4
3
Knowledge at time t:
User interface at time t:
Legend:
Concepts with a communication act
Concepts in the world model
Concepts on the user's interface
Added by the interaction manager module
Added by the fusion module
Added by the user
Corresponding concept
Union operator Plan to remove or hide
Union operator Plan to display

Figure 13.7: *A schematic view of presentation plans in the user adapter and the segment queues in the modality converters*

13.3.3 Language Generator

The language generator works by recognizing a specific communication act and a specific concept-name and substituting a set of properties with their values controlled by the modified AIML format. Figure 13.8 shows some AIML units. The most important elements are listed in Table 13.2. Each category provides some possible templates that can be selected depending on the value of the `<concern>` tag. The same concept and communication act may have many categories, but different set of properties.



```
<aiml>
 <topic name="FIRE">
  <category>
   <dialogActType>statement</dialogActType>
   <concept name="PutFire">
    <attribute name="agent" type="Actor"/>
    <attribute name="beginTime" type="SpaceTime"/>
    <attribute name="location" type="Location"/>
   </concept>
   <template>
    <concern Tv=″<-0.2″ Ta=″>0.2″>
     <random>
      <li>please stay calm, <get name="agent" type="Actor"/>
          <compile conceptName="PutFire" type="VERB"/>
          <get name="getAroundLocation(location)" type="Location"/> since
          <get name="getSinceTime(beginTime)" type="SpaceTime"/>
      </li>
      <li>please remain calm.
          <get name="getAroundLocation(location)" type="Location"/>
          <get name="agent" type="Actor"/> <compile type="TO-BE"/> busy
          <compile conceptName="PutFire" type="VERB-ING"/>
          <get name="getSinceTime(beginTime)" type="SpaceTime"/>
      </li>
     </random>
    </concern>
    …
   </template>
  </category>

  <category>
   <dialogActType>*</dialogActType>
   <concept name="Fireman">
    <attribute name="frequency" type="Frequency"/>
    <attribute name="livingCondition" type="LivingCondition"/>
   </concept>
   <template>
    <concern Tv=″*″ Ta=″*″>
      <get name="frequency" type="Frequency"/>
      <get name="livingCondition" type="LivingCondition"/>
      <compile conceptName="Fireman" type="NOUN"/>
    </concern>
   </template>
  </category>
  …
 </topic>
…
</aiml>
```

Example of resulted text:
"Please remain calm, around Provicialeweg some firemen are busy putting fire out since 15:20 PM."

Figure 13.8: *An example of the text generation from two AIML units (an asterisk "*" means any)*

Figure 13.9 shows the basic algorithm of the language generator. It begins by Identifying the values of the communication act, the corresponding concepts and some minimal context (known attributes of the concepts) within a certain topic, in which the known values appears. A category is selected from an AIML database by ensuring that the most specific known properties match first before any other categories or default (indicated by an asterisk "*"). With known T_v and T_a values that represents the emotional state of the input (see Section 11.4), the language generator randomly selects one of the templates (for this emotional state). It uses the selected template to construct the output text. This component has a parser that processes the template and extracts it into a list of sentences. A sentence has one or more phrases. Inside the

Table 13.2: *Elements of the modified AIML format*

Tag	Description
`<topic>`	Contains current dialogue topic pattern rule.
`<category>`	Marks a "unit of knowledge" in an interaction.
`<dialogAct>`	Contains the communication act that matches to the IM module's selected act.
`<concept>`	Contains the concept name that matches the concept to be explained.
`<properties>`	Contains the property names that are necessary for creating a complete text.
`<template>`	Contains the dialogue text.
`<concern>`	Marks the user's emotional state.

template, there are the `<get>` tag and the `<compile>` tag that can be substituted by a value. These tags are treated as a phrase within a sentence. Using this structure, the language generator can analyze the syntax of the sentence.

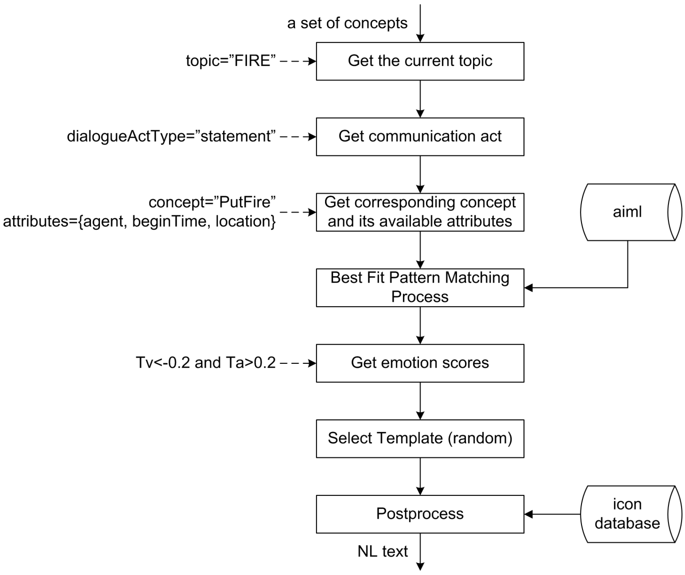


Figure 13.9: *The flowchart of the language generation from a list of inputconcepts*

By knowing the sentence tense from a global variable, in the syntax analysis, a parser applies the cardinality value (plural or singular value) to all phrases in the sentence while substituting the `<get>` tag and the `<compile>` tag by a string value based on their cardinality value. The `<get>` tags can be substituted by:

Another template as the value of the property of a concept. The parser will search the category with the corresponding concept recursively. Using the examples in Figure 13.8, for `<get name = "agent" type = "Actor"/>`, the category of the `Fireman` concept replaces the `<get>` tag in the first unit since the `Fireman` class is a subclass of the `Actor` class. The new found template replaces the `<get>` tag. This parser extracts the template into one or more phrases and adds them into the sentence structure.

A return value of a function. The parser will search the corresponding function and replace the tag by the return value of this function. For example, for the template `<get name = "getAroundLocation(location)" type = "Location")/>`, the function returns "around Provicialeweg" since it finds that the `Location` concept contains information about the address of the `Fireman` concept. This parser assigns the information as the value of the corresponding phrase.

Knowing the cardinality value, the parser in the language generator substitutes the `<compile>` tag using one of the following two values:

The information from the icon database. The parser uses the `rule` information in the icon database (see Section 10.4). One of the sentence rules will be selected based on the current tense of the sentence, the POS of the phrase, and the cardinality of the phrase. For example, for `<compile name = "Fireman" type = "NOUN"/>`, the parser selects a plural form of the noun rule of the `Fireman` concept since the syntax analysis returns `plural` for the category of the `Frequency` concept. This `Frequency` concept explains the property of the `Fireman` concept.

The return value of a function. The parser uses a predefined set of sentence rules. For example, for `<compile name="TO-BE"/>` the parser will search the corresponding POS and apply the cardinality value of the corresponding phrase. This parser replaces the `compile` tag by the value of the POS.

The parser treats the retrieved value of both tags as the value of their corresponding phrase. If the corresponding concept is not available or the corresponding function returns $null$ value, this parser will result an empty string. The processed template will be sent to the modality converter component. The emotional loading of a selected template can be analyzed further using the emotion analyzer component to ensure that this template is the most appropriate to convey the system's reaction to the user's current emotional state.

13.3.4 Visualization Generator

The visualization generator in this research focuses on displaying information in the field of crisis management. A map is utilized to represent the location of a crisis event based on a selected perspective area. A visual language is applied to present information from the world knowledge. Since the input list of linked concepts is built based on the ontology and has a direct link to the icons on the user interface (see Chapter 9), the display of these concepts is considered as one-to-one mapping of the display of their correspondence icon on a certain location on a map. In the map-based display, a function is employed to map world coordinates to screen coordinates. In general, the visualization generator works in the following steps.

1. ***Adding icons.*** The component displays all corresponding icons of active concepts but ensuring none of their instances is displayed more than once.

2. ***Connecting the presented icons.*** The components links all icons based on the relationships between their corresponding concepts. An arrow is used if two icons have a directed relation, such as causality (`causal` or `result`) and possession (`ownBy`). A line is used if two icons have an undirected relation, which means that the concept of one of the icons is a property value of the other.

3. ***Update icons and links.*** The component updates all corresponding icons and their links except those icons that are specified by the user. For such icons and links, only the user him/herself can remove them. This updating action can also be forced by the communication act "removal".

4. ***Highlight concepts or certain locations on the map.*** This action is triggered by a communication act "highlight". It is also used to support visually the information presented by the language component. The highlight is removed after some predefined time or forced by the communication act "removal".

Figure 13.10 shows an example of a visual display on a user's interface. In this example, colors particularly were used to distinguish the user's input (black) from the system's dialogue (green and yellow for concepts resulted by the interpretation process and the visualization generation of a selected communication act, respectively). This display corresponds with the language output example in Figure 13.8.

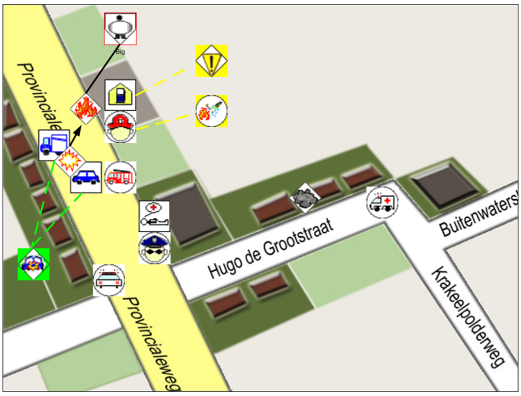


Figure 13.10: *A display showing the situation in a gas station to be dangerous and firemen are busy*

13.3.5 System Design

The class diagram of the `fission` module is presented in Figure 13.11. The `Fission` class of this module receives a selected communication act and a list of linked concepts from the `interactionmanager` module, which define the current knowledge about the user, the task and the world.

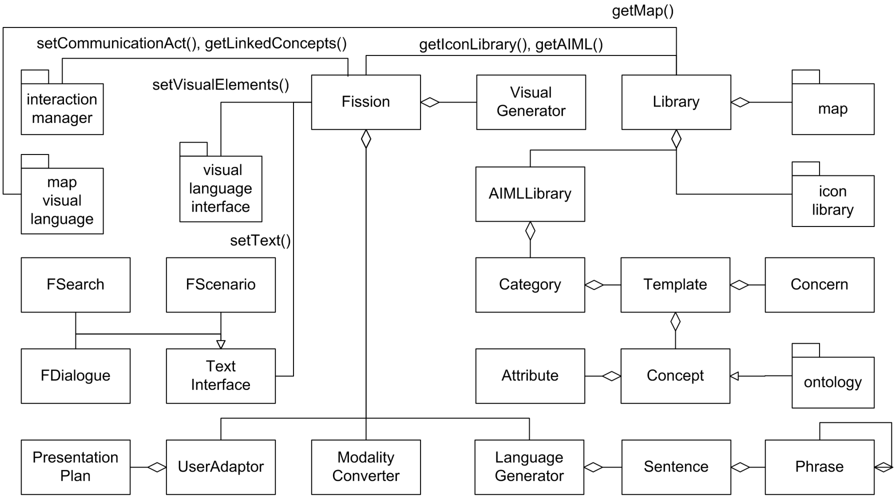


Figure 13.11: *The class diagram of the module for output generation and presentation*

The `UserAdapter` class develops a set `PresentationPlan` objects. These objects are processed in the `ModalityConverter` class to construct a queue of presentation segments and create a full schedule for all available segments at time t. The `LanguageGenerator` class receives a set of concepts and converts them into a set of `Phrase` objects using the `aiml` module and the `iconlibrary` module. For every `PresentationPlan` object, this class works recursively and connects all `Phrase` objects into a set of `Sentence` objects. The `VisualGenerator` class also receives a set of concepts and converts it into a set of linked visual elements. The `AIMLLibrary` class in the `aiml` module contains a list of `Category` objects. The `Category` class has a list of `Template` objects, which are classified based on their `Concern` objects. The `Template` class is defined using one or more `Concept` objects. These objects are defined based on the `ontology` module. The `Attribute` class represents a possible available property of a `Concept` object.

For each `PresentationPlan` object, the `Fission` class sends a set of linked visual symbols and text, respectively, to the `visuallanguageinterface` module and the `TextInterface` class. The messages from the `interactionmanager` module are usually displayed by the `FDialogue` class, except for the communication act "scenario", which is displayed by the `FScenario` class. The `Fission` class also handles user system interaction in general. One of the tasks is for searching a certain icon using a keyword. The results of this task are displayed by the `FSearch` class.

Chapter 14

Case Study: A Multi-User Crisis Observation Interface

In which an AUI that is applied based on the proposed HHI framework is described.

Years after September 11, 2001, efforts to develop technology in crisis management emphasize the development of more sophisticated ICT-based emergency support systems. Our framework proposes a necessary flexible research (test) environment of a HHI system within crisis management, which allows us to change, add or remove devices, modalities, roles and functionalities in a convenient way, without having to spend a lot of time on the reconfiguration of the research environment. An integrated system built based on the framework offers a communication AUI on handheld devices. A wireless ad-hoc architecture connects these devices without a central base station and a blackboard structure is used for distributing information. The infrastructure provides support for rescue teams, the crisis center and civilians, who must work collaboratively. The research reported here focuses on investigating methodologies to improve information acquisition and exchange during crisis respond and rescue activities. It pursues accurate and easy access to information for supporting decision-making, planning and reasoning in crisis situation, and situation awareness of the actors. This includes explorations of the system's ability to adapt and improve its behavior dependent the user's situation over time.

14.1 Reporting Observations

As many computerized services deal with life-death situations and humanity issues, information has the power to save lives possibly many, many lives. People require information to find out what is actually happening and also what they must do to respond to the situations effectively as possible. Crisis response and management involve the collaboration of many people. To perform and coordinate their activities, they must rely on detailed and accurate information about the crisis event, the environment and many more factors. In providing appropriate information to ensure interoperability of emergency services, situation awareness and high-quality care for citizens, the ability to supply information dynamically (such as contextually and temporally correlated) is mandatory. However, current approaches to construct globally consistent views of such an event suffer from problems of (Ramaswamy et al., 2006): (a) the dynamic setting of such events is constantly changing, (b) the information is distributed piecemeal across geographically distant locations, and (c) the complexity of the crisis management organization in general makes it difficult and time consuming to collaborate and verify the obtained information.

The lack of overview is not the only limiting factor in adequate response to crisis situations. Acquisition of detailed and accurate information about the crisis situation is of key importance. Such information can be collected from a variety of sources including observers, rescue workers and sensors. Analysis of past disasters - 9/11 and hurricanes Katrina and Rita by Moore (2006), pointed to communication as a limiting factor in disaster response. Current approaches to coordination of rescue and response activities suffer from the problem that information is neither current nor accurate. Unreliable communication networks, chaotic environments and stressful conditions can make communication during crisis events difficult. The intense nature of crisis situations is believed to result in short term memory loss, confusion, difficulties in setting priorities and decision-making (Farberow & Frederick, 1978). These result in fragmented and badly structured communication or even leaving out some relevant data (Sharma et al., 2003).

Recent developments in technology offer citizens possibilities for communication using handheld devices, such as PDAs and smart phones. Recently, police departments, firefighters and paramedics started showing interest in utilized handheld devices as part of toolkits for exchanging information with control rooms, headquarters and hospitals quickly and efficiently (IBM.com, 1999; MESA-Project, 2001). As mobility becomes ubiquitous, multimodality becomes the inherent basis of the interaction paradigm. Multimodal user interfaces designed for multi-device scenarios can offer various ways for users to interact in a more natural fashion with provided services (Oviatt et al., 2004). The introduction of novel ICT in the crisis management domain can help to provide more detailed and accurate situation overviews that are current and shared amongst all management levels (Moore, 2006). For example, web-based reporting interfaces in CAMAS (Mehrotra et al., 2004) and VCMC (Otten et al., 2004) allow users to send reports using natural language messages. For control rooms, Sharma et al. (2003) developed a multimodal framework to facilitate decision making. Furthermore, Tatomir & Rothkrantz (2006) has developed an visual language-based interface for sharing and merging icon-based topological maps in damaged buildings. The knowledge can be used for reasoning about the state of the building and providing guidance to given locations.

14.2 Application Concept

A novelty proposed by the communication interface developed in this research is that human's observation-based context awareness. These observations are context sensitive. They are based on multimodal input in a given context and may be affected by emotional state and mood due to the intense nature of the crisis. Figure 14.1 shows the conceptual model of the (mobile) reporting observation interface.

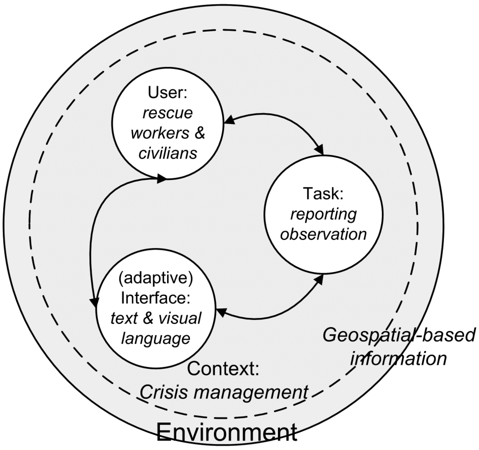


Figure 14.1: *The model of mobile use context of the reporting observation interface*

In the process of reporting observations, the information may become ambiguous, incomplete and language dependent. The interface offers a standard representation to facilitate the exchange of information, to promote universal understanding, to adequately address the communication of mission critical information across different disciplines and cultures, and to provide a common representation for communication needs. The common representation consists of the various information objects, their interrelationship and their potential relevance in crisis situations, whose meanings agreed upon by multiple actors (who are working collaboratively in resolving crisis).

The crisis situation is the context of the interaction. Handheld devices are the main communication device, which may have different types of computation platforms and different constraints of input and output modalities and capabilities. At any time, a user is associated with a location in the geographical space and communicates with others using his/her handheld. Providing fast and easy interactions is the goal of the interface design concept. The essence of the proposed concept is multimodal, context-aware and affective interaction. Knowledge of the user, the task and the world is built up based on both initialization and interaction with the user.

The interface offers compact and simple navigation. The main window was designed to support main interactions and equipped with features that are necessary for creating messages and for accessing information. To retrieve and deliver context sensitive information, the interface provides an observation map. As pointed out by Dymon (2003), crisis management often relies on teams of people who must collaboratively derive knowledge from geospatial information. The map allows users to attach information relevant to a particular location of a crisis event. Visual symbols are used to represent objects, events and relations that involved in the crisis situation. Such symbols can also represent links to more detailed or other forms of information.

14.3 System Design

The design concept of the interface supports the users to create messages as fast as possible. The elements that are used more frequently for creating messages are displayed around the input area. Therefore, this could increase the user's attention for the coinciding action-perception space. At the same time, distinctive visual cues are provided. Therefore, users do not have to devote their full attention to operate the application. In this case, only active elements of the interface can be selected or activated by the user. To support faster interaction, two adaptability approaches are applied: (1) multimodal user interaction and (2) context-sensitive knowledge content. Both approaches include the analysis of the emotional aspect of the user.

Multimodal User Interaction

Text and visual language are the main communication form on the interface (Figure 14.2). Two options for the input interface are offered: (1) the adaptive Cirrin interface and (2) the free visual language-based interface. These two interfaces are linked by two types of visual symbols (see Figure 10.5) from a map-based interface. These (sub) interfaces are provided to allow the users to determine which of the communication methods is the most appropriate for a given situation. The adaptability user interaction is provided by five means: (a) the frequent character prompting, (b) the word completion and syntax correction on the adaptive Cirrin, (c) the icon prediction on the visual language-based interface, (d) the visual discrimination for a selected icon from the rest of unselected icons, and (e) the content, service and option maintenance for different modalities and communication platforms.

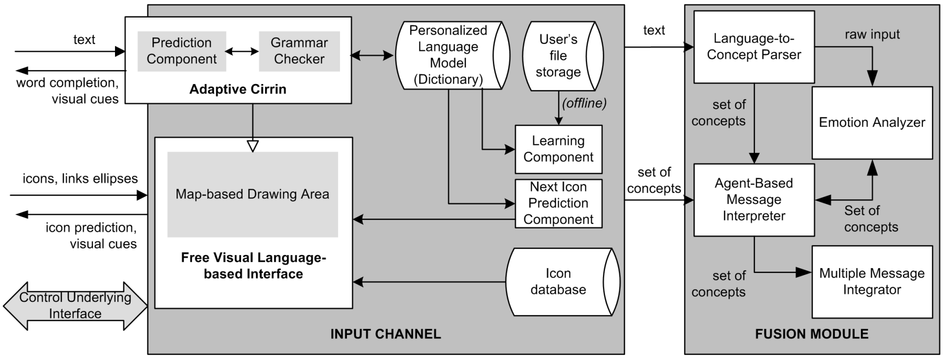

text
word completion, visual cues
Prediction Component
Grammar Checker
Adaptive Cirrin
Personalized Language Model (Dictionary)
User's file storage
(offline)
Learning Component
Next Icon Prediction Component
icons, links ellipses
icon prediction, visual cues
Map-based Drawing Area
Free Visual Language-based Interface
Icon database
Control Underlying Interface
INPUT CHANNEL
text
set of concepts
Language-to-Concept Parser
raw input
Emotion Analyzer
set of concepts
Agent-Based Message Interpreter
Set of concepts
set of concepts
Multiple Message Integrator
FUSION MODULE

Figure 14.2: *A schematic view of the architecture of the input modules*

The communication interface produces five output options: (1) the interpretation of the user's message in a form of crisis scenario; (2) the resulted typed text (in the adaptive Cirrin interface); (3) the dialogue text of user system interaction; (4) the corresponding visual symbols on the user's map as representations of the updated knowledge of the world (in a multi-user environment); and (5) the feedback underlying interaction on the interface (such as visual cues on the user's selection).

Context-Sensitive Knowledge Content

Figure 14.3 shows developed modules that handle: (1) interpreting and merging incoming information, (2) managing user system interaction and (3) specifying and producing context-sensitive (and user-tailored) information - by the employment of scripts and ontology from and to multimodal, multiple users and multiple devices. The latest includes allocating and coordinating information across available media, such as typed or spoken language, visual language, and graphics.

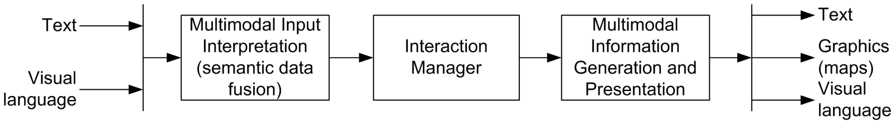

Text
Visual language
Multimodal Input Interpretation (semantic data fusion)
Interaction Manager
Multimodal Information Generation and Presentation
Text
Graphics (maps)
Visual language

Figure 14.3: *Modules that are involved in user system interaction*

The interface was designed to support various roles within the crisis management, including professionals and civilians in the field and control room. A centralized Fusion Manager integrates every newly reported situation from all users and adapts the knowledge accordingly then sends it back to the network and shares with the users (Figure 14.4).

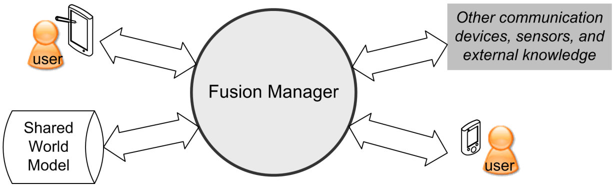

user
Fusion Manager
Other communication devices, sensors, and external knowledge
Shared World Model
user

Figure 14.4: *A schematic overview of the communication system: multimodal, multi-user and multi-device*

Figure 14.5 shows a schematic view of the interpretation processes in the single-user interface and the (multi-user) Fusion Manager. Ontology and scripts are utilized in all processes. The fusion module in a single-user interface builds concepts from various preprocessed input modalities, which are supplied by all the available input recognition modules. The results include interpretation of the user's emotional state. They are handed over to the IM module and may assist this module in forming feedback to the user. The selection of appropriate feedback highly depends on the current emotional state of the user. The IM module sends the user's processed message to the Fusion Manager. The Fusion Manager attempts to relate the existing structure of the current world model with the new message and then broadcasts the latest world knowledge. Together with a selected communication act from the IM module, the fission module of a single user interface adapts the current world model and produces a user-tailored information presentation. Here, the presentation of information depends on the user's perspective, emotional state and modalities choice. The integrated reports are considered as the most up to date knowledge of the world, which are displayed on the user's map.

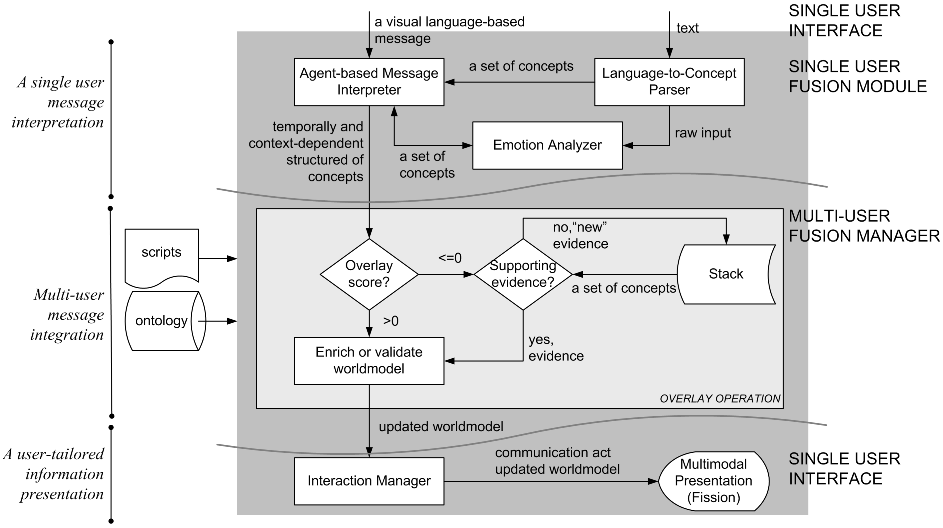

a visual language-based message
text
SINGLE USER INTERFACE
A single user message interpretation
Agent-based Message Interpreter
a set of concepts
Language-to-Concept Parser
SINGLE USER FUSION MODULE
temporally and context-dependent structured of concepts
a set of concepts
Emotion Analyzer
raw input
MULTI-USER FUSION MANAGER
no,"new" evidence
scripts
Overlay score?
<=0
Supporting evidence?
a set of concepts
Stack
Multi-user message integration
ontology
>0
Enrich or validate worldmodel
yes, evidence
OVERLAY OPERATION
updated worldmodel
A user-tailored information presentation
Interaction Manager
communication act updated worldmodel
Multimodal Presentation (Fission)
SINGLE USER INTERFACE

Figure 14.5: *A schematic flowchart of proposed approaches in context awareness computation*

14.3.1 Setting

The following settings illustrate the use of the reporting observation interface.

The First Theory. *It is an ordinary bright morning. The traffic is quite busy. People go to work or school. Tim is walking on a pedestrian path. A sudden squeal-sound of tires and steel against the asphalt comes from a distant behind. It is followed by loud banging, crashing, sounding of car horns and screaming. Everyone is looking to the direction with a great shock. Tim sees long traffic jams around the intersection. He takes an initiative to report the situation using his reporting observation tool. Tim describes the traffic jams and a high possibility of traffic congestions.*

Wrong Hypothesis. *Tim runs to the intersection. He sees crowded people around a certain object laying on a sidewalk. It looks like a female posture. Tim suspects her being hit by a car and these people are helping her. He thinks that she should be handled by a paramedic as soon as possible. Therefore, he sends a new report by placing an icon "victim" on the correct location. Tim also puts an icon "accident" near the intersection* *on the map interface. He draws a link between the two icons. After the message is sent, Tim receives a feedback. It seems that other people have reported about some events a bit distance from the intersection. He looks around whether the situation fits with the incoming information. Few meters from an intersection, he sees a car is wrecked underneath a truck. Other cars tried to avoid collision with each other. This causes traffic jams on both side of the road. Tim finds that his input was wrong. He deletes his inputted "accident" on the map interface. Automatically, the link is also removed.*

***Emotional Intense**. Latter, Tim finds out that the female person is not the victim rather she was shocked witnessing such an accident. He sees that the driver is still inside the wrecked car. People are trying to help him out but need a tool to break through the car. The car's passenger is injured and wounded. These victims need help soon. He describes the situation as detail and accurate as possible using words, icons, links and ellipses on his reporting observation interface. He types as many situational phrases as he can think, such as "badly injured", "awful smell", "very long traffic" and so on. The interface returns feedback that he must remain calm because the policemen are on their way, while paramedics just departed from the nearest hospital.*

***Intelligent Conclusion**. Policemen come in cars and on motorcycles. At the same time, Tim receives new information on his interface. He compares it with the real life situation. Apparently, there is another car behind the truck and hidden from his sight. This car is flipped over. The driver is still inside unconscious. From the interface, Tim finds out that the situation is extremely dangerous. Moreover, at the side of the road, there is a gas station. Explosions and fire may be inevitable. He shouts to the crowd to move back. Right at the same moment, the policemen also receive the same conclusion on their communication device. The policemen immediately block off the road and evacuate people to the other side of the road. Some of them clear the road and get the traffic moving from the other side of the road and behind the scene. Two policemen try to open the flipped car and help the driver out. Another policeman secures the truck driver to aside. When the policemen and the driver are about few meters away from the car, suddenly it explodes. They are thrown into the air and badly injured.*

***Observing from Afar**. Everyone stays back, nobody dares to come. The explosion causes fire in the area. The car and the truck are smashed and burnt. At the side of the road, the gas station is also half burnt. Two ambulances come to scene. The paramedics help and evacuate the victims to the safe area. Fire-trucks are coming. The firemen start busy putting out the fire around the area, while the policemen clear the area. Tim sees that some officers use their communication device to report the situation. He also frequently checks his reporting observation interface. Tim updates his description.*

***Language Independent**. The ambulances leave the scene, another comes. People are pushed away more to the side. Both sides of the road are starting to be empty, except those wrecked vehicles, blackened burnt. Tim can see black smoke coming from the gas station. Next to the place where he is standing, there is an Asian man. This man is also busy with his handheld device. Tim immediately notices that this person* *has the same reporting interface as his. However, Tim does not recognize the language coming from this person's device and the dialogue text also displays strange characters. But, the visual symbols arrangements on the map interface are exactly the same as on Tim's interface. After using the interface several times, Tim becomes quite handy in creating messages, both text and visual language. He can understand fairly the meaning conveyed by an arrangement of visual symbols without reading the resulted interpreted scenario. Therefore, by looking at the interface of the person next to him, Tim is also able to recognize rather fast the message it conveys, compare it with his interface and check whether he misses any information.*

14.3.2 Task Analysis

Figure 14.6 displays the scheme of reporting observation tasks. It is assumed that with the current technology in the Global Positioning Service, an application can run certain routines to detect the position of the user anywhere in the world. Using the result, the developed interface can show approximately a map where the user is located.

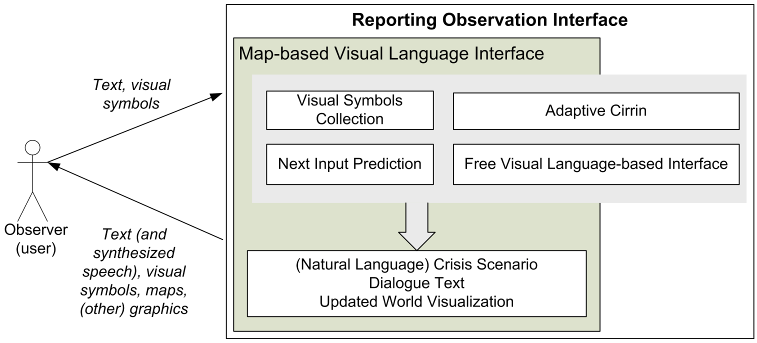

Reporting Observation Interface
Map-based Visual Language Interface
Visual Symbols Collection
Adaptive Cirrin
Next Input Prediction
Free Visual Language-based Interface
(Natural Language) Crisis Scenario
Dialogue Text
Updated World Visualization
Text, visual symbols
Observer (user)
Text (and synthesized speech), visual symbols, maps, (other) graphics

Figure 14.6: *Reporting observation scheme - illustrates the main interaction by the actor*

Before interacting with the icons, the user has to choose the context of what he wants to speak about. For example, about crisis events, involved actors, transportations, attributes of the concepts and so on. The interface will display the icons that are categorized for these concepts in the icon menu. The user can select any icon from the icon menu and place it on a map interface.

The user may create non-spatial messages using text or visual language by selecting one of the icons that represents these types of messages and placing it on the map interface. The location of these icons is interpreted as the spatial relation of the message in the real world. Links and groupings of these icons to other icons represent the relation of the message (or the main concept in the message) with other concepts. A single click on these types of icons (on the map interface) means selecting the icon, double clicks means creating or editing the message.

Each time a visual symbol has been selected, next possible icons will be calculated, predicted and ranked. The user can select an icon from the calculation results, which is much faster than searching it in the menu.

Table 14.1: *User tasks*

No.	Task
1.	Select a context from the list of groups of icons from the icon menu.
2.	Select an icon from the icon menu (from a selected group of concepts).
3.	Select an icon that represents a non-spatial input interface.
4.	Select an input from the next input prediction.
5.	Search an icon based on a keyword.
6.	Place a new icon.
7.	Link two icons.
8.	Group some icons.
9.	Delete a visual symbol.
10.	Send the observation.
11.	View the resulted scenario.
12.	Create, edit and save a non-spatial information .
13.	Change, zoom and pan the map.
14.	Activate a TTS synthesis for the resulted scenario.
15.	Change output language.

Table 14.1 shows a summary of the user tasks derived from the design concept and the scenarios. The 1-11 user tasks were selected for the implementation of the first prototype interface to enable meaningful assessment.

14.3.3 Interface Design

The user interacts with the interface mainly via a touch screen using a pen (or a finger) as the only interaction element. Figure 14.7 shows the prototype interface of the reporting observation interface. The following is the description of the main interface.

Toolbar. The toolbar consists of operational icons (see Figure 10.5), which are grouped based on their functionality. Visual cues are displayed to distinct those icons that can be selected during interaction. The toolbar includes icons for creating non-spatial messages. Selection of such icons can be done by clicking one of icons then directly select a location in the drawing area. These icons are only available on the main window.

Icon Menu. In the icon menu, icons are grouped based on their concept. An interaction element similar to *a tree-based tabular* holds the hierarchy of the icons. On the display the icons comprise two levels, however, actually they consist of three levels. The top level presents the concept icon and incorporate compound icons. Querying any concept will initiate a move to the second level, which contains base icons

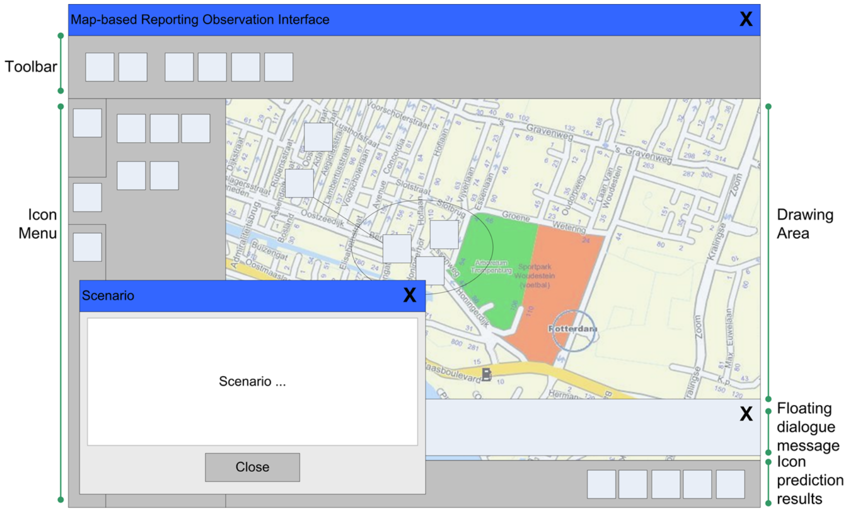


Figure 14.7: *Low fidelity prototype of the reporting observation main window*

grouped into sub-concepts. An icon can be queried by two ways: (a) select an icon in the icon menu, then drag and drop it into the drawing area and (b) select an icon in the icon menu, then select a location in the drawing area.

Drawing Area. The drawing area is a workspace where a user can arrange visual symbols to represent his/her message. A deletion action will remove only one visual symbol. A user can select any icon in the drawing area using the pointer then press the "delete" icon in the toolbar to remove it. On the same drawing area, visual feedback (the system's dialogue concepts) is displayed in forms of visual symbols.

Textual Output. A floating dialogue displays the system's dialogue. It can be a question, a request or a statement depending on the selected communication act from the IM module. Viewing the crisis scenario from the current report can be done by selecting the "view" option. A separate window is displayed. The main window can be accessed again after other sub-windows are closed. A similar procedure works on the free visual language-based interface (Figure 14.8(b)). On the adaptive Cirrin interface (Figure 14.8(a)), the text output is produced from selections of characters.

Icon Prediction. The icon prediction gives a list of suggestions for the next possible icons. They are calculated and ranked based on n-grams language model of user selection (see Section 4.4). The user can select any icon from the suggestion list and append it into the drawing area. This feature is also offered in the free visual language-based interface. In the adaptive Cirrin, the frequent character key prompting and the word completion are applied based on the user input characters (see Chapter 5).

Speech Presentation. The dialogue text will be read aloud by a TTS synthesizer. For the resulted crisis scenario and the text output, this TTS synthesizer can be activated

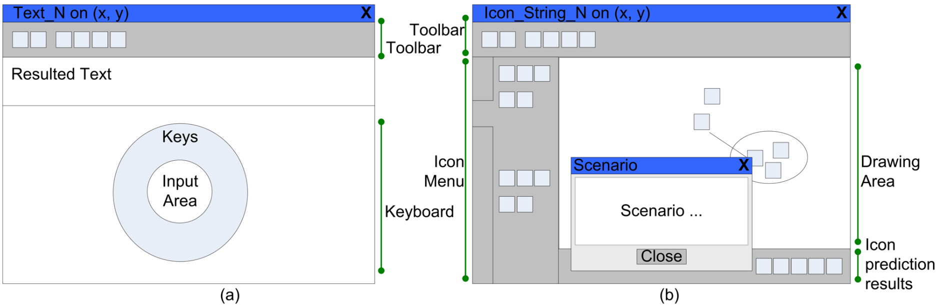


Figure 14.8: *Non-spatial input interfaces: (a) adaptive Cirrin interface and (b) free visual language-based interface*

using the "speak" icon in the toolbar.

14.4 Implementation

Recent crisis events have shown that existing communication infrastructures can become overloaded or even breakdown. The need for technology to cope with non-deterministic environments resulting from the global wired communication breakdown has never been more apparent. Therefore, we designed the communication infrastructure between users using a distributed-system architecture based on a Mobile Ad-hoc Network (MANET) that connects their communication devices (Figure 14.9). The architecture allows a peer-to-peer wireless network that transmits from a handheld to a handheld without the use of a central base station (Klapwijk, 2005).

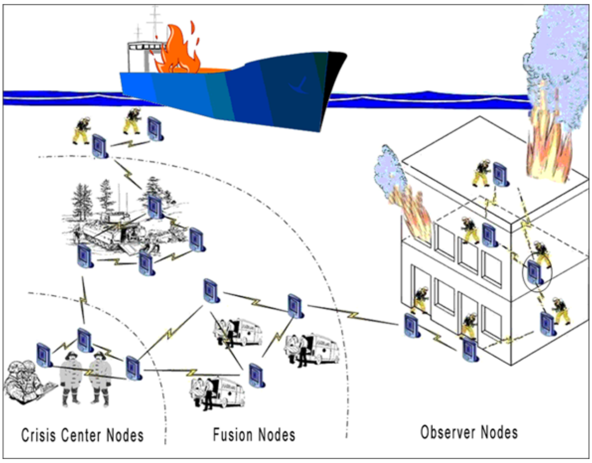


Figure 14.9: *Visualization: the mobile ad-hoc network (MANET)*

All modules are integrated using the iROS middleware system (Johanson et al., 2002). The modules communicate with each other through a common event heap (Fitrianie et al., 2007). Modules send events containing XML messages to the heap and other modules can subscribe to receive certain types of messages from the heap. One of the main advantages of this communication infrastructure is the fact that modules can be connected in an ad hoc fashion. As the system was designed to deal with multiple users with a range of different devices and network connections, the fusion manager enhances the robustness and flexibility of the core iROS system. In a multi-user system, the same module can exist in different instances for different users. Messages are addressed to a specific user or be broadcasted to all users. A user is addressed using an unique user ID. When a user's device or module temporarily become unavailable, for example when an unreliable (wireless) network link goes down, the manager adds an error recovery layer to iROS, which distinguishes essential messages and streaming messages. The latter ones are only relevant at a specific time and lose importance as soon as a new message is produced in the stream. They are therefore not subject to error recovery. However, the manager does ensure that essential messages are delivered to all intended recipients.

The reporting observation interface was implemented under `Java SE 2.0`. All knowledge is developed in the form of XML and RDF/OWL. The current version of the interface provides English as the main language. Translation to other languages has not yet implemented. The conversion of RDF/OWL to `Java 2.0` is performed using `Protege 3.2.1`. The following presents the UML design of the implemented system from technical point of view: context model, static model and dynamic model.

Context Model

The use case diagram below (Figure 14.10) shows the tasks and the main actor. Table 14.2-14.4 describe the use cases.

Table 14.2: *Use case description*

Name	Description
Input a character	Description: Typing a character on the adaptive Cirrin interface; Actor: User; Pre-condition: The adaptive Cirrin interface is open; Exception: None; Result: The new character is appended on the input area and the text area; the word completion and the visual cues of keys are displayed.
Select an icon	Description: Placing or selecting an icon on a drawing area; Actor: User; Pre-condition: A visual language-based interface is open; Exception: None; Result: The new icon is in the drawing area and highlighted; a list of next icon suggestions is displayed.
Draw an ellipse	Description: Creating an ellipse around one or more icons; Actor: User; Pre-condition: At least one icon is in the drawing area; Exception: None; Result: An ellipse is drawn following the movement of the pen input.

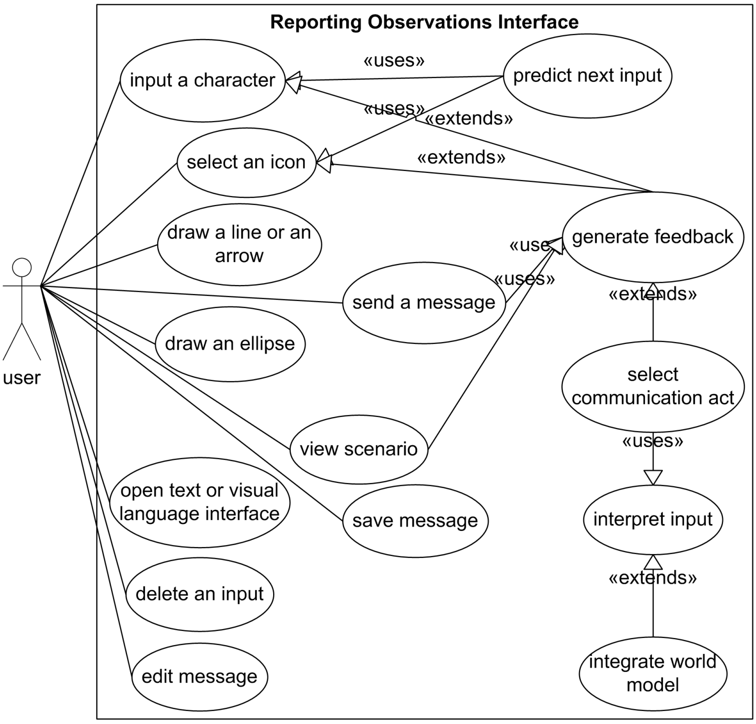


Figure 14.10: *Use case diagram of the reporting observation interface*

Table 14.3: *Cont'd: Use case description - 1*

Name	Description
Draw a line or an arrow	Description: Creating a line or an arrow between two icons; Actor: User; Pre-condition: At least two icons are in the drawing area; Exception: There is already a link between the selected icons; Result: A link is drawn between the two icons and highlighted.
Open a text or a visual language-based interface	Description: Double clicks on a non-spatial message-based icon; Actor: User; Pre-condition: Selecting a non-spatial message-based icon in the toolbar and placing it into the map interface; Exception: None; Result: The corresponding input interface is displayed.
Edit message	Description: Displaying the corresponding non-spatial message on the corresponding interface; Actor: User; Pre-condition: Double clicks on a non-spatial message-based icon on the map interface; Exception: One click action will only cause the icon is selected and highlighted; Result: The corresponding interface and message are displayed.

Table 14.4: *Cont'd: Use case description - 2*

Name	Description
Delete an input	Description: Deleting a visual symbol in the drawing area; Actor: User; Pre-condition: At least one selected visual symbol in the drawing area; Exception: None; Result: The visual symbol is removed from the drawing area; if the symbol is an icon and has a link to another icon, this link will be removed too; if the icon is the only element that intersects with an ellipse, this ellipse will be removed too; a new set of icon suggestions is displayed.
Save message	Description: Saving a non-spatial message; Actor: User; Pre-condition: A non-spatial message-based interface is open; Exception: None; Result: The message is stored.
Send a message	Description: Sending a message to the network; Actor: User; Pre-condition: At least one icon is on the map interface; Exception: None; Result: The message is sent and updated knowledge is received.
View the scenario	Description: Displaying the interpreted scenario; Actor: User; Pre-condition: At least one icon is on the map interface; Exception: None; Result: The input message is interpreted and the result is displayed.
Predict next input	Description: Predicting next inputs based on syntax rules and n-gram language model; Actor: None; Pre-condition: At least one icon or one character is in the input; Exception: None; Result: A new word completion and visual cues of keys are displayed on the adaptive Cirrin interface, a set of next icon suggestions is displayed.
Interpret an input	Description: Interpreting the inputted messages; Actor: None; Pre-condition: The option "send" or "view" is selected; Exception: None; Result: World model based on the current message is created.
Integrate world model	Description: Updating the world knowledge with the new (sent) message; Actor: None; Pre-condition: Receiving a new message from a user; Exception: None; Result: Knowledge of the world is updated.
Select a communication act	Description: Selecting a communication act from a list of communication strategies based on knowledge of the user, the task and the world; Actor: None; Pre-condition: The option "send" or "view" is selected; Exception: None; Result: A communication act is selected and sent to the output module.
Generate feedback	Description: Producing output, which can be: a crisis scenario, a resulted text, a dialogue text or an updated map display; Actor: None; Pre-condition: A communication act is selected; Exception: The resulted text (on the adaptive Cirrin interface) is produced directly by the corresponding input module; Result: Output is displayed (synchronously).

Static Model

Figure 14.11 shows the component diagram of the interface. Table 14.5-14.6 describes all components in the diagram and Table 14.7 describes the library.

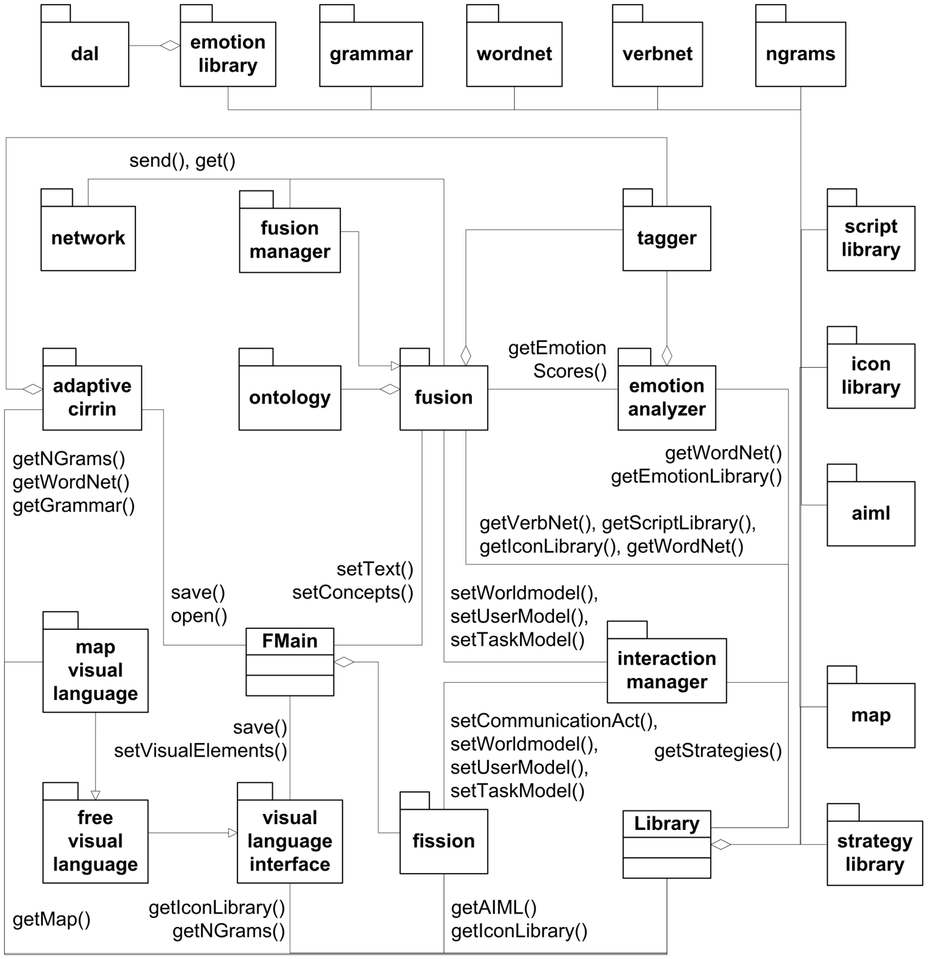


Figure 14.11: *The component diagram of the reporting observation interface*

Table 14.5: *Component description*

Name	Description
`network` module	Handles the communication between the user interface and iROS.
`FMain` class	Handles the interaction between the `fission` module and the I/O.
`adaptivecirrin` module	Handles user interaction on the adaptive Cirrin (see Figure 5.4).

Table 14.6: *Cont'd: Component description*

Name	Description
`visuallanguageinterface` module	Handles user interaction on a visual language-based interface in general.
`freevisuallanguage` module	Handles user interaction on the free visual language-based interface (see Figure 8.6).
`ontology` module	Handles the crisis management ontology (see Figure 9.6).
`mapvisuallanguage` module	Handles user interaction on a map display based on the free visual language.
`emotionanalyzer` module	Analyzes the user's emotional state and the urgency of the situation based on the user input (see Figure 11.8).
`tagger` module	Handles the string extraction into a set of POS using Qtag POS tagger (Tufiş & Mason, 1998).
`fusion` module	Handles the interpretation of the input (see Figure 12.9).
`fusionmanager` module	Handles the integration of a user's world view with the system's knowledge of the world.
`interactionmanager` module	Selects appropriate communication acts (see Figure 13.4).
`fission` module	Produces output based on the selected communication act and the knowledge about the user, the task and the world (see Figure 13.11).
`Library` class	Handles the xml-based internal inputs needed by other modules.

Table 14.7: *Components in the Library module*

Name	Description
`scriptlibrary` module	Store scripts (see Figure 9.14).
`iconlibrary` module	Store icons and their headings (see Figure 10.7).
`aiml` module	Store AIML units (see Figure 13.11).
`map` module	Store maps and their geographical information.
`strategylibrary` module	Stores the communication strategies (see Figure 13.4).
`emotionlibrary` module	Handles the `dal` module (see Figure 11.8).
`grammar` module	Stores production rules for English grammar.
`verbNet` module	Handles `VerbNet v.3.1`.
`wordnet` module	Handles `WordNet v.2.1`.
`ngrams` module	Handles n-grams-based personalized dictionary.

Dynamic Model

Figure 14.12 is the transition diagram of the use cases "input character", "select an icon", "view scenario", and "send message". The `freevisuallanguage` module has a `DrawingArea` class, which contains icons, links and ellipses. When the user adds a new visual symbol to the `DrawingArea` class or a character on an `InputArea` object (in the `adaptivecirrin` module), the class that handles prediction, calculates and ranks the next possible inputs using the n-gram-based user personalized dictionary from the `Library` class. The adaptive Cirrin and the free visual language-based interfaces can only be accessed from the map-based interface.

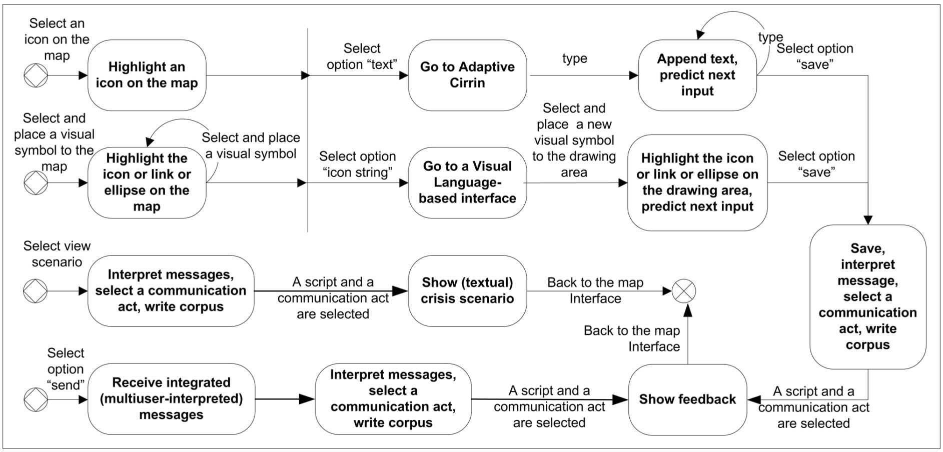

Select an icon on the map
Highlight an icon on the map
Select option "text"
Go to Adaptive Cirrin
type
Append text, predict next input
type
Select option "save"
Select and place a visual symbol to the map
Highlight the icon or link or ellipse on the map
Select and place a visual symbol
Select option "icon string"
Go to a Visual Language-based interface
Select and place a new visual symbol to the drawing area
Highlight the icon or link or ellipse on the drawing area, predict next input
Select option "save"
Save, interpret message, select a communication act, write corpus
Select view scenario
Interpret messages, select a communication act, write corpus
A script and a communication act are selected
Show (textual) crisis scenario
Back to the map Interface
Back to the map Interface
Select option "send"
Receive integrated (multiuser-interpreted) messages
Interpret messages, select a communication act, write corpus
A script and a communication act are selected
Show feedback
A script and a communication act are selected

Figure 14.12: *Transaction diagram of the use cases "input character", "select an icon", "view scenario" and "send message"*

When the user selects the "save" option on any non-spatial-based interface, the message will be stored. On a visual language-based interface, the user has to select the "view" (scenario) option to see the interpretation of the input. Selecting the options "save", "view" or "send" activates classes in the `fusion` module, which interprets the new messages, and classes in the `fission` module, which produces the feedback. The `fusionmanager` module has classes that integrate the new message with the global world knowledge via the network, which is stored by classes in the `ontology` module. To produce the feedback (including the scenario), the classes in the `fission` module need the current user model, task model and world model from the `ontology` module and a selected communication act from classes in the `interactionmanager` module. The latest module selects communication acts using classes in the `strategylibrary` module from the `Library` class.

14.5 User Testing

An experiment has been conducted consisting a small-scale user test in a laboratory setting. It was aimed at performing a test on the design concept and addressing usability issues of an AUI that was built based on the proposed HHI framework. With the implemented reporting observation interface, we set up the experiment to answer two questions: (1) can the users create messages to convey their concepts in mind using modalities? and (2) what can help the users to construct richer messages?

Since the objective of the development of the AUI was not meant to replace common ways of communication, such as using telephone, SMS or e-mails, we did not compare our communication interface with any text or speech-based communication interfaces. In this experiment, we tried to reduce the effect of learning the communication interface as a new system, in user performance by providing a short manual and asking all participants to interact with the interface before taking the actual test.

14.5.1 Methodology

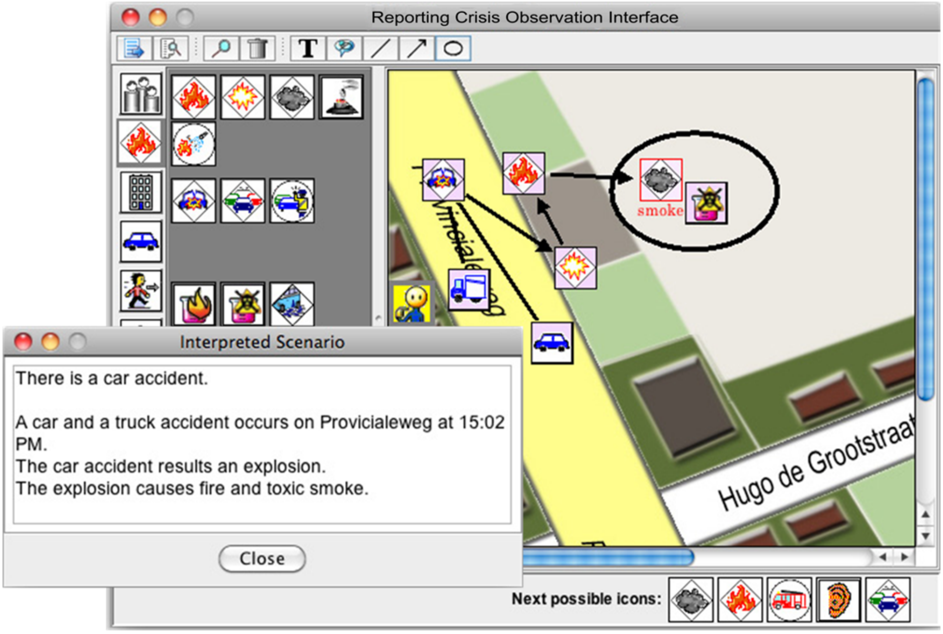
Unfortunately, the reporting crisis observation interface could not be tested in a real crisis, because it was difficult to create a controlled experiment of a disaster just to test the system. Therefore, this preliminary experiment had to be done in simulated crisis situations. We performed the experiment using a small room with the tasks that were created real for witnesses during crisis events. The thinking aloud method (Boren & Ramey, 2000) was applied to allow a large qualitative results from a small number of participants. Using this protocol, the participants were encouraged by the experimenter to verbalized their thought during the experiment. A more natural conversation situation was expected for motivating the participants to give more information.

Test Material. The experiment used: (1) a PC with a large monitor and a speaker, (2) a tablet PC with a pen and (3) an experimenter a participant task booklets. Only one experimenter assisted all participants during the experiment. The user interactions were logged and the user's speech during the experiment was recorded. The logger program is able to unobtrusively gather click stream data on every specified task that has been completed by the user.

This experiment was divided into three sessions: (1) trial, (2) test and (3) questionnaire. The tasks were composed to give our participants freedom to answer. There was not incorrect answer, except if the participant did not answer the question at all. These tasks were also created in such ways that the participants can experience five aspects of usability issues of an AUI on a handheld device: (1) input design, (2) feedback design, (3) layout design, (4) navigation structures, and (5) affordance of the controls.

During the experiment, the participants were equipped with a combination of a map-display and a free language-based interface (Figure 14.13). The interface was set with the configuration in Table 14.8. In addition, the online learning component was not available. In this way, the language model had the same training samples for all sessions and all participants. To be able to focus on the assessment of the free visual language-based interface, an options for creating additional messages using the adaptive Cirrin interface was disabled too.

Figure 14.14 shows the guide window that led the participants through the trial and test sessions. By clicking the available check boxes sequentially, the participants followed all cases in both sessions. The participants used a booklet to write down



There is a car accident.

A car and a truck accident occurs on Provicialeweg at 15:02 PM.
The car accident results an explosion.
The explosion causes fire and toxic smoke.

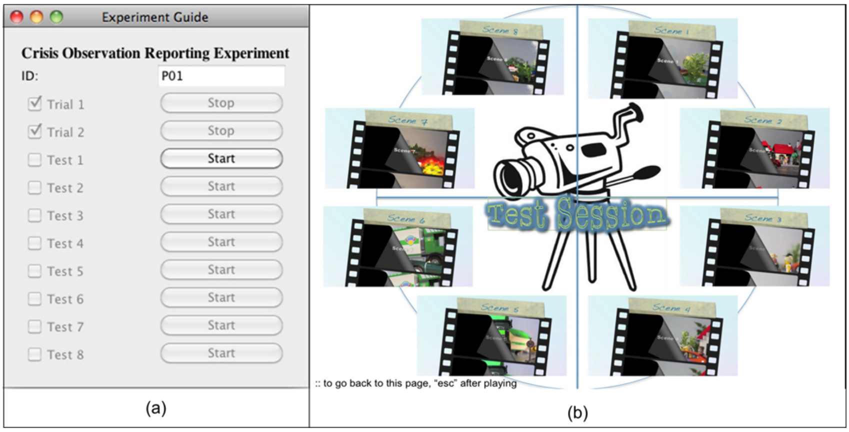
Figure 14.13: *The Reporting Crisis Observation Interface*

Table 14.8: *The configuration of the reporting observation interface in the user testing*

Parameter	Configuration
Window size	440*640 pixels (landscape).
Ontology size	127 classes.
Icon database size	110 icons classified into 10 groups.
Script library size	47 scripts about car accident, fire and explosion.
Prediction library size	121 unigrams, 740 bigrams and 534 trigrams in the field of crisis management.
Number of AIML units	544 categories classified into 11 topics in the field of crisis management.
Strategy library size	134 strategy units.
Emotion library size	200 keywords in the field of crisis management, 85 aggressive words and 28 qualifiers. The `dal` module contains 8742 words in a 2D space with bipolar dimension of valence and arousal.

the initial observation of given crisis situations and the textual version of their visual language-based messages (on the provided user interface) after each task. This booklet was also used for filling the questionnaire.

Test Execution. The initial approach to the experiment was to try out the inter-



Experiment Guide

Crisis Observation Reporting Experiment

ID: P01

☑ Trial 1 Stop
☑ Trial 2 Stop
□ Test 1 Start
□ Test 2 Start
□ Test 3 Start
□ Test 4 Start
□ Test 5 Start
□ Test 6 Start
□ Test 7 Start
□ Test 8 Start

:: to go back to this page, "esc" after playing

Figure 14.14: *(a) The guide window and (b) eight video illustrations of a crisis situation*

face. By selecting the trial session on the guide window, the corresponding interface was displayed automatically by the guide. For each case, the participants were asked to describe two illustrations of crisis situations (using pictures on the large monitor - similar to Figure 7.17) using the developed interface. For each illustration, the participants were asked to write down their initial observation (before the reporting task) and the textual version of their message (they wanted to convey - after the reporting task) on their booklet. During this session, we expected the participants to familiarize themselves with the interface.

The real experiment began by selecting the test session on the guide window. Here, the participants were asked to describe eight illustrations of crisis situations. Instead of using pictures, in this session, we used videos that were created using a sequence of pictures with sound to depict crisis events. These illustrations represent a sequence of scenes of crisis scenario (Figure 14.14(b)). Appendix A.1 presents the detailed scenario including some screenshots of the video. The setting is an illustration of a real life situation built using Lego-bricks. Figure 14.15 shows examples of these scenes. The flow of the scenes was designed so that the participants could feel the increasing tension of the crisis, which implied more dangerous situations. During the test, the experimenter played each illustration video on the large monitor. By following and describing the sequence, we expected the participants would have the mental idea of what was happening. Similar to the trial session, after each illustration, the participants were asked to write down their initial observations. Using the guide window, these test users worked on each specified task. They used the interface provided for reporting their observations. The participants were asked to describe the observations as much as possible and afterward, to write down the textual version of their message.

A simulated crisis center was implemented to send messages about the incoming

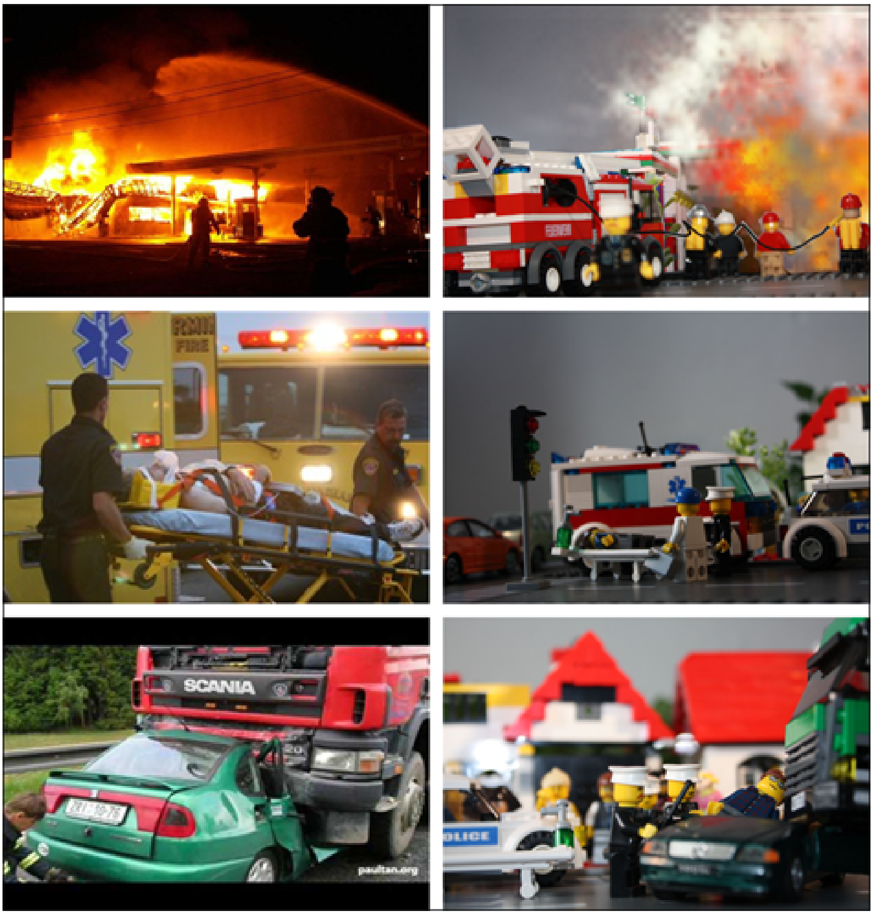


Figure 14.15: *Using Lego-bricks to illustrate real life situations: the work of firemen, paramedics, and policemen*

of the rescue workers ((Benjamins, 2006)). A timer was set to define when a message was sent to the `fusionmanager` module.

Questionnaire. At the end of the experiment, the participants were asked to fill in a questionnaire. The questionnaire focused on eight aspects: (1) whether it is fast and easy to find and select the intended icons, (2) whether the grouping of icons makes their selection faster and less error, (3) whether the visual symbols are sufficient for conveying the intended message, (4) whether it is easy to make correction, (5) whether the user understands how to construct a message, (6) whether the prediction and the search options make the input faster, (7) whether the constructed scenario can convey the intended message, and (8) whether the user has the awareness of what was happening during the reporting. The questionnaire contained 30 statements. The participants were asked to give rate 1-5 for each statement, such as 1: strongly disagree, 2: disagree, 3: neutral, 4: agree, and 5: strongly agree.

Participants. Eight people took part in the experiment and one person as a pilot. All participants were asked to do all tasks in all sessions in the same sequential order. In this way, we expected that these participants had the same starting and progressing knowledge about the interface in general. We assumed that all participants had the same level on computer knowledge and experience using a pen as input device. There was no assumption on distinction of age, gender or cultural differences.

14.5.2 Measurement

From the experimental results, the following measurements were taken:

1. ***Report completeness***. This aspects was measured by analyzing reported concepts and their relations. Initially, a list of concepts and relations that might occur in every test scene was created by the developer. This list was considered as the minimal reports of the simulated situations to achieve context awareness. Further, the (textual) initial observation and the textual version of the user messages were transcribed from the participant booklet into a set of concepts using the `LanguageToConcept` class in the `fusion` module. The next was available concepts and their relations in the transcription results and the interpreted messages created using the reporting interface for each test scene were listed. Finally, the results were compared to each other.

2. ***Number of errors during performing tasks***. This aspect was measured by calculating the number of delete actions in performing a task.

3. ***Lostness in "icon space"***. To measure disorientation or the sense of being 'lost' in a visual language-based interface, we used the formula of measurement of the sense of being lost in hypermedia (Smith, 1996):

   $$L = \sqrt{(\frac{N}{S-1})^2 + (\frac{R}{N-1})^2}$$

   where L is a lostness rating, S is total number of nodes visited whilst searching, N is number of different nodes visited whilst searching and R number of nodes which need to be visited to complete a task. We expected that since the icon space also was made up of interlinked icons and all information was displayed only by icons, the visual language-based interface also might give cognitive overload and disorientation to its users. The problem refers to "users cannot find what they are looking for". The ideal path to have R was calculated different depending on the messages that created by each participant. The visited "nodes" referred to the visited concepts (selecting). For a perfect search in the icon space $L = 0$. Increased lostness rating could be viewed in terms of degradation of user performance.

4. ***The emotion loading of the inputted messages.*** This aspects was measured by analyzing the calculation results of the `emotionanalysis` module.

5. ***Usability of the interface.*** This aspect was measured using a dialog scheme. The transcription from this experiment was coded into six schemes: (1) [$MAZES$], corresponding to meaningless token, such as: "uh", "oh", "hmm" and the like; (2) $EXPLORE$, corresponding to when the participant tried to understand the interface; (3) $READ$, corresponding to when the participant tried to understand an icon; (4) $EXPLAIN$, corresponding to when the participant told the reason of his/her action; (5) $OPINION$, corresponding to when the participant told his/her opinion or feeling about the interface; and (6) $OTHER$, corresponding to other than above dialog code. To find more about usability aspects on the communication interface, we analyzed three inputs: (1) the transcriptions result of $EXPLORE$, $READ$, $OPINION$, and $OTHER$, (2) the questionnaire and (3) the extra remarks that we explicitly asked the participants to write on their booklet.

14.5.3 Result Analysis

The result of the study are discussed in the following four parts: message constructions, situation awareness, usability assessment and user satisfaction.

Message Constructions

All participants accomplished their tasks with relevant answers. Most of the time, the participants chose the icons, arranged them in the drawing area, and related them using lines, arrows, and ellipses confidently. Only about 15% of the total tasks, these participants had doubts to choose either a line or an arrow to connect two concepts. Similar situation occurred when they had to choose either using an ellipse or just placing the icons closer to each other (if there were more than two icons). We found that the participants used lines and arrows almost interchangeably except for three relations: (1) temporal "this happens before that", (2) causal "this happens because that", and (3) owned "this is owned by that". To describe these relations, in particular, the test users always used arrows "this → that". The ellipses were mainly used to inform that "one or more relations may exist among these icons". Two participants mentioned that they used the ellipses to distinguish one group of related icons from the others. Overall, there was not indication the need of other types of lines or shapes to represent relations between concepts, which implied that such visual symbols provided in the experiment were sufficient in the domain in focus.

In average, the participants used about eight icons to convey their message. The minimum number of icons that were used to construct a message was three and the maximum was 19; while the minimum number of relations between icons after the interpretation process was one and the maximum was 11. Only 25% of the total task,

Table 14.9: *The Coverage of input concepts and their relations for test no. 1 - 8*

Corpus	Average #Missing Concepts	Average #Missing Relations	Average #Subs. w/ Avail. Concepts	Average #New Concepts	Average #New Relations
A: Initial Observation	1.78 ± 0.67	2.23 ± 1.11	0.21 ± 0.87	3.11 ± 1.41	0.81 ± 0.43
B: Report on Interface	4.22 ± 1.31	4.56 ± 2.32	3.11 ± 1.21		
C: Textual Ver. Report	3.40 ± 1.25	3.94 ± 2.68	2.01 ± 0.35	1.54 ± 0.86	2.22 ± 1.32
$A \cup B \cup C$	1.38 ± 0.05	0.37 ± 0.05	0.25 ± 0.25	3.11 ± 1.41	2.22 ± 1.43
$A \cap B \cap C$	4.22 ± 1.31	4.56 ± 2.32	3.11 ± 1.21		

the participants used verb icons in the message limited to "hear", "see", "go", and "come". They used more noun icons and adjective icons (to explain the nouns).

Table 14.9 shows the average missing concepts and relations after the comparison. Directly after watching the scenes of the crisis, the test users had relatively the same state of mind as the developer. However, during message constructions, the results indicated that the participants adapted the concepts in mind with the provided vocabulary in the icon menu. In some cases, it appeared that the participants did not realized that the interface actually provided the intended concept. Other cases, these participants just took the first concept that they considered, could replace the intended concept in mind. About half of the missing concepts were substituted using other concepts. For example the message "a lot of people" instead using the provided "a lot" concept, most participants put more than one "people" icon on the map interface. Therefore, the average substituting missing concepts with available concepts (provided by the interface) is higher for the report on the interface than in the initial observation. Our test users tried to correct their decision (to explain what they actually wanted to convey) on the textual version of their reports.

The high number of the missing concepts occurred mostly in the early test session (see Figure 14.16), when our participants tried to familiarize the interface. Two participants informed us that after a while they could form a mental idea what kind of concepts they would used in the message while watching the video. Nevertheless, we found that in some cases, the test users still could not find a concept that could fit with their messages. This was also one of the reasons of the other half missing concepts. Figure 14.17 shows that the high number of unsuccessful searching action of the desired concept occurred mostly in the early test session. During this time, one participant even decided to select an icon by scanning icons in the menu one by one.

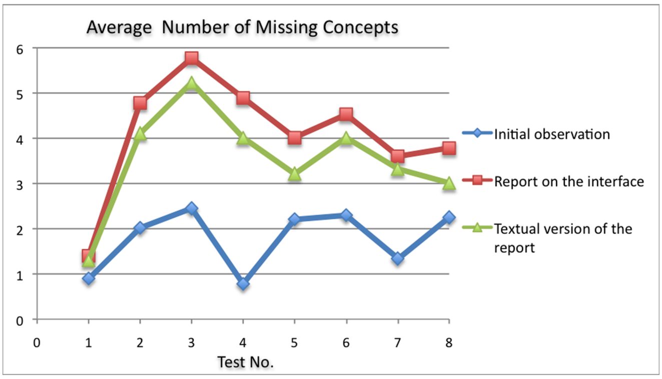


Figure 14.16: *The average number of missing concepts in test no. 1-8*

Other participants were appeared to keep searching the desired concept if it was considered important for describing a situation.

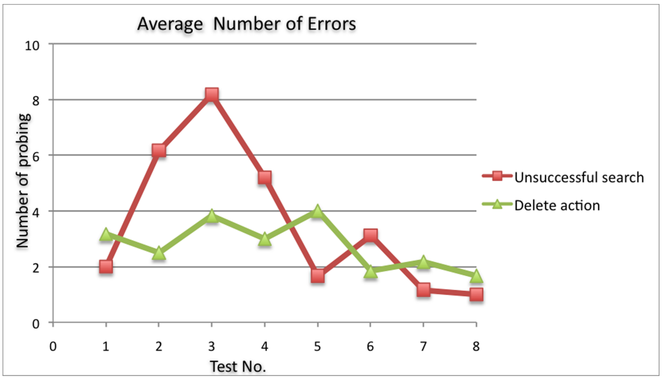


Figure 14.17: *The average of unsuccessful searching and delete actions in test no. 1-8*

The results also showed a low number of delete actions because the participants always checked the `tip-text` of the icons before selection. Most of these actions were by mistake when they wanted to delete the shapes. This made these participants to re-search the deleted icon. Furthermore, the deleting actions also occurred when the participants did not agree with the resulted scenario. Although most of the participants still checked the scenario, after a while, they decided to ignore if the result was not what they intended. They used the participant booklet, instead, to explain their intention. Together, these problems influenced the lostness rating (Figure 14.18). However, if we look at the tendency of the curve, our test participants indicated to be able to adapt themselves with the limitations.

The experiment results showed that some concepts and relations in the textual version of user reports were different from the initial observation. One participant explained that he thought he had to adapt his textual report to what he might be able to report on the interface. This included the consideration of what kind icons could represent their message. On the other hand, the initial observation was considered as a description of the level of understanding of what was happening. Table 14.9 shows that the initial observation had more new concepts than the textual report.

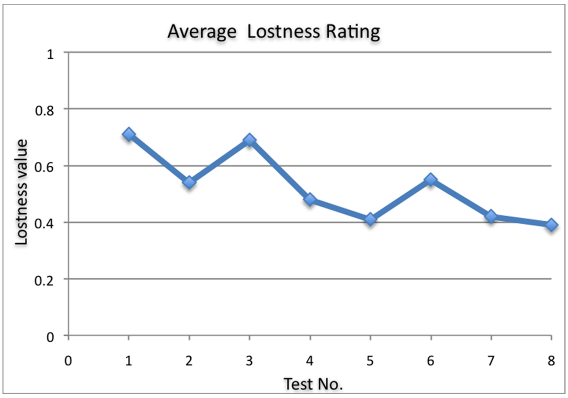


Figure 14.18: *The lostness rating for test no. 1-8*

Situation Awareness

The fusion module received all concepts from the input messages. However, it failed to connect about 20% of relations between these concepts, due to:

- The intended relations have not been implemented. For example, when the user linked the "car accident" icon and the "injured person" icon with an arrow for indicating a causal relation, the agents in the interpretation algorithm would try to find this inverse property first before other related properties. However, these agents only found it in the `CarAccident` concept, but not in the `Actor` concept.

- If there were more than one eligible related icons, the agents always chose the closer one. This policy made other concepts that were connected by shapes but placed further was not chosen. It is appeared that the relationships between two concepts using shapes preceded the possible relationships based on their distance. The participants found it inconvenient especially when many icons have been placed in a small location of the map.

Because of the second reason, the module interpreted a relation incorrectly when an icon was placed on a certain object of a map, such as a building and a street. The current algorithm of input interpretation included knowledge of static objects on the map. Due to this knowledge, for example, when a "car accident" icon was placed on a certain street on the map, a relationship was established between the CarAccident concept and the Street concept that yielded the location of the event. However, this kind of relation also occurred when a "female person" icon and a "house" icon were placed closed to each other. The interpretation module automatically assigned a relation between them and resulted "the house of the female person ...". This made the participants confused because they did not expect to find such a result.

The experimental results showed that the inference engine of the fusion module was able to select appropriate scripts based on the input concepts. However, the engine did not infer any new concept in the user workspace because the input messages always contained key concepts. It appeared that the participants always viewed the scenario or sent the message after they thought their message was complete. This indicated that the missing concepts in Table 14.9 were considered as non-key concepts.

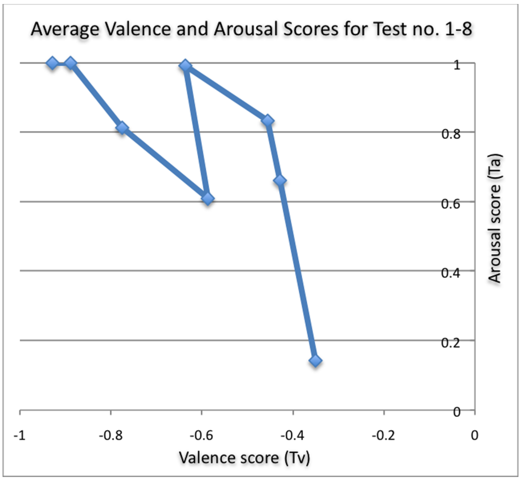


Figure 14.19: *The average results of the emotion analyzer for test no. 1-8*

Figure 14.19 shows the average results of the `emotionanalyzer` module. This module was able to capture the emotion loading of the input. It was able to detect the worsening of the situation. For example, in test no. 4, the participants reported about car accidents, injured victims, traffic jams, ambulances and so on. The emotion was less negative and less active when the participants reported only the work of the policemen and the paramedics than when the reports contained explosions and fire. At this moment, the negative emotion was in higher intensity.

The experimental results showed that the `fusionmanager` module was able to receive the user's concepts from the IM module and integrate them with the messages coming from the simulated crisis center. The integrated world model was broadcasted and accepted by the IM module on the user's communication device. The IM module was able to select a communication act based on these updated knowledge and the observed user's emotion. It triggered the fission module to produce output based on the selected communication act. When the interface informed about the current situation from the crisis center's perspective, the participants notified these messages and expected new events would occur in the next test scenes. The biggest drawback was the resulted scenario still could not convey the user message because:

- The current `aiml` module is not yet expressive enough to take care different structures of the relationships between concepts due to some relations between concepts and many combinations of known properties of concepts were still not yet covered.

- Error in the interpretation results.

Apart from reducing the trust of the users toward the interpretation module of the interface, it was not possible to draw any conclusion whether this problem could reduce the user's situation awareness. However, if we look at the coverage of the union of the input concepts in Table 14.9, the fact that the input concepts almost covered the concepts and relations of the developer might indicate that the participants had the same situation awareness as what the developer expected.

Usability Assessment

Almost all remarks of the participants pointed into the same direction. The most important points are summarized in the following.

- *Icon Design.* There was not any significant complaint on the icon design. The participants mentioned that they could understand the meaning of icons. Some of these understanding came after they checked the `tip-text` of the icons. This `tip-text` was very useful at the beginning of the session, especially for indexical icons. However, displaying `tip-text` action influenced the lostness rating because the test users tended to check them one by one. Therefore, in future, this action can be an optional setting for novice users, which can be disabled when the users become an expert to make the drawing area less crowded.

  Most participants found it was easy to find and select intended icons in the icon menu. However, these participants mentioned that they could not understand the meaning of the frame of icons since their meaning did not change because of their frame. Moreover, in the icon menu, these icons were not grouped based on their frame. From the developer's view, it was easier to design an icon inside a square frame than other frames since this frame has more space. The experimental results yielded future work for finding a better icon classification method.

Although the `tip-text` was not available for the toolbar icons, the participants could recognize them without any problem. This was due to the fact that most of them were popular used icons.

- *Icon Menu Design.* Most participants could understand the concept of grouping icons. They mentioned that they realized this after some interactions with the interface. For some icons, the participants could not interpret the individual alone. The image might not indicate what the icon represented or its `tip-text` was not sufficient to indicate how the icon can be used to form a message. Instead, the icon's related group would give the idea of the category of the icon. However, we found that the test user had problems with verb icons. These participants found it was overwhelming to search a verb icon among many verb icons. The participants could not understand the sub-grouping of these icons in the second layer. They advised us to group this type of icons based on their functionalities. For example, verbs that are used for rescue and responding activities, for communicating with other, for transferring an item and so on.

  The test users found the current arrangement of the first layer was helpful when they searched a certain icon. The current design arranged the icons in such a way it looked sequential. It started with the icons that represented crisis events, then continued with the icons that represented objects involved in the events, and ended with the attributes of these events and objects.

- *Layout Design.* Most participants found that the functionality of most elements in the interface were easy to use and to learn. These participants could understand the functionality of the icon menu, the prediction results and the toolbar because they were grouped and placed in different parts of the interface. It appeared that about 50% of our test users did not use the prediction results in four first test cases. It seemed they forgot about this facility since their attention was occupied with inspecting icons in the icon menu. In the last four test cases, these participants started using it more often. Almost two times per task in average, they took benefit from the prediction to create their message. These test users mentioned that from the prediction results, most of the time at least one icon could be used in their message.

- *Navigation Structure.* The participants found that the design concept of the interface was straightforward. The interface provided sufficient means for the task they had to perform. These participants also could understand the changing appearance of the visual symbols as the hint of their selection. They could explain to us that in a pen-based interaction, the icons should have been selected first for displaying their `tip-text` (not hovering as in a mouse-based interaction).

  The test users could understand quite quickly that they could only use a "select-drag-and-drop" interaction for the icon menu and the prediction results. These users also understood that there were several ways to interact with the icons in the toolbar. For example, a "click" for the options "send", "view" and "search"; a "click-and-click" for the "delete" option, and a "click-press-and-drag" for the options drawing "line", "arrow" and "ellipse".

- *Affordance in Control.* The participants found it was inconvenient to create messages as soon as there were many icons on the map display. Too crowded visual display made them disorientated. In some cases, the icons had to put in stack. Some participants re-checked each icon in their message several times to make sure they were sufficiently conveying the message. However, it was difficult to find the previous selected icons, even more difficult to make corrections and to assign any relation to these icons. This made the participants did unnecessary delete actions (Figure 14.17). Two particular participants used multiple random ellipses to indicate "these icons are related". The other two users advised us to provide layers for dividing their message into sub-messages. Other participants asked for zooming the map, which had not yet been implemented.

  It appeared, in the first two test cases, most participants were influenced by the resulted scenario. These participants thought they arranged their icons incorrectly. As mentioned before, this occurred especially when there were too many icons placed in the same area of the map. After a while, the participants ignored the results and, in exchange, used the booklet to explain their messages. In addition, three participants advised us to use a separate window where they could view and assign attributes to the icons on the map display. There participants had a computer science background and we considered them as expert users.

  These problems influenced the lostness rating.

User Satisfaction

In general, our participants were enthusiastic. The test users liked the idea of representing messages by arranging icons and some shapes. Most participants found the interface design had met their expectation, the vocabulary was acceptable for the current domain in focus, and the visual symbols were sufficient. However, our test user advised some improvements, such as: (1) a more suitable method for classifying icons; (2) a more usable presentation of prediction results; (3) providing a way on separating a message into sub-messages, therefore, the interface will not be too crowded; (4) providing a better way on attaching attributes of concepts that represent objects and events; and (5) presenting a more accurate resulted crisis scenario to convey the input message.

Chapter 15

Conclusion and Recommendations

In which the work on developing a HHI framework of an AUI that is multimodal, context-aware and affective, on handheld devices, is concluded. Some recommendations for future work are presented.

Living in a mobile connected world opens up numerous opportunities for both work and leisure activities and makes our daily conduct both more efficient and more exciting. These opportunities, however, pose challenges for the technical and HCI communities. Mobility is most often considered as an attribute of a computing device or a user in general, not as an attribute of a user and a device during the interaction. In this thesis, we focused precisely on the latter and showed solutions for interacting with a mobile computing device, in particular a (mobile) handheld device, in arbitrary location and situation. The research has achieved the objectives by going through a complete design cycle and finding interesting results on the framework for HHI. Our work made preliminary steps in proposing a methodology for developing adaptive user interfaces (AUI) on handheld devices. This methodology has combined and applied interesting concepts, such as alternative input designs, knowledge engineering, context-aware analysis, interaction management, information presentation and applications for handheld devices. A prototype system as our research demonstrator developed based on the framework has proven the concepts.

This chapter summarizes the main findings of our research and draws their implications on the research into developing AUIs on handheld devices. It thereby pertains to answer every formulated research questions. The experimental studies, the development of the prototype system and the user test results underline each of the answers as conclusion of the research work. After highlighting these findings, the chapter poses some directions for future developments of the framework of HHI (in general) and of AUIs on handheld devices (in particular).

1. ***What is the state of the art in user system interaction on handheld devices?***

Typical problems in mobile user system interaction are mainly due to the small size of the devices and the mobile use, which heavily affects the design and usability of user interfaces for interaction and presentation. The problems present additional requirements on developing user interfaces for handheld devices, which include: (a) the need for faster interaction, (b) the need for alternative input options, (c) the need for methods to display sufficient information on a small screen and (d) the need to allow the users to devote less than their full attention during operating such a device.

To cover all these requirements, an exploration of an AUI for handheld devices has led to the development of a framework for a multimodal, context-aware and affective HHI. Communication using handheld devices was the focus domain of the research. A system that is built based on this framework is able to autonomously adjust its display and available actions to current goals, abilities and the emotional state of the user by assessing the status of the user, the interaction task and the environment.

Our research combines models, methods and approaches in artificial intelligence, software engineering and psychology. It includes range of fields, such as communication, HCI, mobile user interfaces, AUI, multimodal systems, usability, ontology, knowledge engineering, database, context awareness, NLP, visual language, reasoning and graphics. This research has covered the investigation on natural input options and output presentation to maintain interactive systems on handheld devices. In line with these, we also carried out literature study, software development and experiments on knowledge construction and maintenance about users, systems and contexts to support both adaptive input and output.

Since context plays an important role in interpreting user messages and presenting information, the developed framework provides mechanisms to fuse multiple input modalities into a context dependent interpretation of the current situation of the particular user, the interaction task and the environment. The interpretation result is used to update the system belief about the user, the task and the environment. With constraints defined by the obtained knowledge, the feedback generation modules of the framework are able to generate multimodal dynamic, coherent and synchronized multimodal representation. The constraints include the user's emotional state and the urgency of the information.

As proof of the proposed concepts, a prototype system based on the framework has been developed and is used as a supporting system for crisis management. It is a comprehensive experimental system for reporting observations using handheld devices in a MANET-based communication. The communication occurs using the combination of text, visual symbols and graphics on map-based interfaces. The system is equipped with the ability to interpret the message, create a world model and develop a crisis scenario. The design of the system includes direct feedback to user inputs, allowing for verification and alternation of the information and ways for collaborating information.

2. ***How can we improve the performance of user input?***

Our research has proposed two alternative input options: (1) a circular keyboard (for text input) and (2) a visual language-based input. Both of them provide ways for fast input and error recovery. They are accelerated with predictive and language-based features. These input designs have a learning component that is able to learn the ways of user inputting information and the context of his/her messages.

An experiment has been conducted in selecting an input prediction method that performs on ambiguous inputs with a small dictionary. The experiment compared three language models: hash tables, n-gram, and Markov model, underlying the design of a telephone pad and the input task on a public transportation information service. The results showed that both n-gram and Markov model work better than the hash tables even with a small dictionary. However, to build the Markov model, unigram and bigram data are needed. This experiment led to the development of the icon prediction using an n-gram model. The user test results showed that the users took advantage of the prediction to accelerate the searching time for an intended icon. The developed icon prediction system was able to suggest a list of next possible icons using relatively little n-gram data. The data was collected from user selections during interaction with a visual language-based interface. However, the test results also indicated that the usefulness of the prediction depends strongly on the way the suggestion list is presented on the interface. This yields future work for the design of this interface. One of the ideas is to apply a dynamic interactive display by displaying the prediction results underneath a selected icon in the drawing area. This list of suggested icons moves along the position of the selected icon.

The second experiment assessed whether or not an adaptive input is necessary (especially on handheld devices) by comparing the most common English words taken from the BNC with personal datasets, such as personal documents, e-mails and chat logs. Although the BNC covers most of the personal corpus, the experimental results showed that the intersection of the personal datasets is small. Moreover, the word completion showed better performance using a relatively small dictionary containing the highest frequency words based on normal word usage. These indicated that besides personal word usage, the ability to improve effective text entry and typing rate might also be dependent on the context of the user task. A personalized task-based dictionary can offer a more efficient prediction than a large common dictionary. The results can also imply to the accuracy of the prediction if syntactically implausible words are also excluded from the prediction space. In this way, besides saving time and energy in inputting, a text entry interface can also assist the users in the composition of well-formed text. In addition, the number of user inputs for a desired word can be reduced if the interface takes an assumption that a suggested word is rejected after the user selects the next character. Therefore, the user can have a better language coverage since each suggested word is shown only once.

Learning from previous research on alternative input options for handheld devices, the adaptive Cirrin has been developed for single-handed use, mainly using a pen input (or a finger). This on-screen keyboard was designed based on the Cirrin layout of (Mankoff & Abowd, 1998). The user test results showed that at least three advantages of the adaptive Cirrin over the Qwerty layout: (1) the combination of the tapping-based and the motion-based text input, (2) the visual cues by expanding the key's size, which help the user to select and find characters fast and easily, and (3) less possibility to select the neighbor by mistake, since the cursor is mostly coming from the inside of the annulus. The language-based features, such as a personalized and adaptive task-based dictionary, frequent character prompting, word completion, and a grammar checker with suffix completion, accelerated the users' text entry and improved their performance. We expect the users will need a small period of time to learn the Cirrin layout and perform better using the adaptive Cirrin than the on-screen Qwerty. Future work has to be done in an improved design of the space key and the fisheye style option.

On a visual language-based interface, users can create messages using an arrangement of visual symbols as realization of their concepts or ideas in mind. The use of icons to represent concepts or ideas makes user interaction particularly suitable for a fast interaction across user diversity in language-independent contexts. The metamorphic development of the visual language began on its first prototype application, in which the message can be created using a sequential arrangement of icons. This proposal was improved by the second prototype, in which messages can be created using a 2D acyclic-graph of icons, where the predecessor explains the successor icons. The 2D sentences can capture much richer message than the sequential visual language-based sentences. Methods in NLP, such as Xtag (Joshi & Vijay-Shanker, 1999) and VerbNet (Schuler, 2005), are combined to analyze the syntax and semantics of visual language sentences and convert them to text/speech simultaneously. This approach naturally and elegantly captures the interaction between the graph structure of the visual sentences and the tree syntax structure, and the inferential interactions between multiple possible meanings generated by the sentences. The experimental results indicated that the users need a small period of time to learn the language.

We were inspired by the way (younger) children make drawings to represent their visual thinking and people sketch to help disambiguate what they are saying. Here, the research was aimed at a free way of creating visual language-based messages using a spatial arrangement of graphics symbols, such as icons, lines, arrows and ellipses. The order and position in which visual symbols are arranged cannot be determined beforehand, since there is not any specific grammar controlling the arrangement. The user test results indicated that the users were able to naturally and confidently arrange visual symbols on a drawing area to represent their concepts and ideas in mind. It appeared that the constructed messages were able to represent the cognitive mind of the users at a particular moment. The results also showed that the visual symbols provided to represent relations between concepts were sufficient to convey the user message. However, the experimental results also showed that displaying multiple visual symbols on a small screen causes its limitations to be quickly reached. The interface looked crowded, the icons were displayed in stacks and it was difficult to verify and compose messages. Future work needs to be done on dealing with this problem. Two possible ways can be explored as a solution. The first is by separating a message into sub-messages or layers. The second is by providing a general view of concepts in the drawing area and an option to zoom in to a specific view of the drawing area. It is also advisable to provide different interaction styles that can accommodate both novice and expert users. For example, expert users may demand a more elaborate way to relate concepts in the drawing area than novices. In another case, novices may rely more on `tip-text` of icons, while expert users may find it unnecessary and only making the interface looks more crowded. Additionally, more research can be done to increase the expressiveness and flexibility of the arrangement of provided visual symbols to cover more relations between concepts.

The proposed visual language paradigm is not meant for replacing any primary communication, such as text and speech, but to open up new possibilities for fast communication, interaction and obtaining information. Both developed linguistic-based and (grammar-) free visual languages have their own pros and cons. The former visual language is more robust since it is based on profound theories, however, its linguistic grammar still restricts the users to following linguistic lines behind sentence constructions. On the other hand, the free visual language allows a more natural way of arranging visual symbols to create a message, however, more research in methods for interpreting the meaning underlying the spatial and temporal arrangement of symbols, is necessary. Moreover, the current development of the visual language-based interface is dedicated for direct communication in a specific context, that is the field of crisis management; the interface was designed to contain a limited number of icons that represent concepts in the domain of interest. In addition, some aspects of language, such as verb tenses, have not been deeply developed. Problems in infinite searching for an icon that is suitable for a certain concept or creating an unintended message out of a particular arrangement of icons may be inevitable due to bad icon designs, the lack of vocabulary or a bad method of displaying the icons. More experiments in real life situations are necessary for gathering more user requirements and determining the performance of the language and the interface that employs this language.

Finally, a study in developing an icon database particularly has led to three different views of icon design considerations:

- *From the handheld device technology view.* Loading icons (up to about 100 primitives) is fast if they are stored as vector data. The modification of this type of data using indirect drawing (via IMR) uses less processing power than direct drawing.

- *From the usability view.* Using popularly used icons, such as road signs, software interface icons and institution icons, is more usable. However, their meaning should not be changed. The users can better recognize iconic and indexical icons than symbolic icons. Movement visual cues can be used for verbs, propositions and adjectives. Simple images with a small number of elements are easier for recognition. Moreover, an understandable concept or group of icons leads to understandable icons, that is based on their category, such as events, transportations, colors. In addition, it is advisable to group indexical-verb icons apart from iconic icons.

- *From the interface view.* The interpretation results of an arrangement of visual symbols should be provided directly. Therefore, the users can learn the icons in trials. Dealing with a large database of icons, a visual language-based interface should support a faster interaction, such as utilizing a search engine, an icon prediction, plausible syntax hints and distinctive cues of syntactically plausible icons. In this way, the interface can offer reliable interpretation of inputted messages.

In a long-term usage of a visual language-based interface, it is advisable to allow users to add and edit icon vocabulary. However, these operations should be designed in a way that the consistency of the database is still maintained.

3. ***How can we represent the domain knowledge of a (mobile-handheld-) HCI?***

To represent a logical/abstract view about the user, the task and the world, an ontology is selected for the knowledge representation. The knowledge representation enables input interpretation to build context awareness using the emerging relationships between features and concepts in the ontology. The interpretation process is controlled by a hybrid agent-based architecture, which applies decentralized rules and strategies for identifying concepts and the message structure in the user workspace. It involves the mechanism of building coherent, context dependent semantic structures from dynamic concepts of the communicated messages. A dynamic script approach was utilized to identify basic conditions that are held in the workspace. These conditions are characterized by a unique set of concepts from the ontology. The interpretation results of multiple inputs from a particular user are then validated and integrated using an overlay operation. This operation can also be used for anaphora analysis based on previous input messages from a particular user and integrating input messages from multiple users.

The development of the system built based on the framework has demonstrated that the model and structure of the developed knowledge representation have fulfill its main purpose for supporting intercommunication between components in the system and the context-aware requirements. The use of ontology and scripts as knowledge representation makes inferences easier to verify afterwards. All knowledge may be executed in a different order, but they can be designed and specified sequentially in the order of the conditions that are held in the domain of interest. As a drawback, for the system to be fully expressive, all possible conditions have to be converted into sets of concepts, scripts and icons in advance since communicating messages about conditions that have not been specified, is not possible. Moreover, the current knowledge is still dedicated for a specific domain, that is the field of crisis management. Therefore, more corpus and expert-based knowledge engineering in different domains are necessary to allow the framework to be developed into a widespread application in different domain. Additionally, further research could be done in generating new scripts for conditions by learning.

The test results showed that the agents of the input interpretation module were able to process all concepts in the user workspace using the ontology. These agents were able to build groups of related concepts, build relations between concepts, activate new concepts and fill in properties of the active concepts. The current approach into the activation of these relationships focuses on applying the law of proximity for interpreting how people relate certain concepts and on using the scripts to identify the whole form of these related concepts. The inference engine of the interpretation module was able to select appropriate scripts that were identified by the active concepts and their assigned relationships. However, from the test results, it appeared that the users consciously decided on a certain way (placing the related concepts near to each other or using a certain shape). Solely based on the proximity was not yet sufficient to interpret how the users how people arrange concepts in the workspace. The law offers methods to identify the relationships between pre-clustered concepts based on where the user placed them in the workspace, but fails when the clusters cannot be defined, for example when the interface is too full or the concepts are placed too sparse. Moreover, it was indicated that the relationships between two concepts using shapes preceded the possible relationships based on their distance. A deeper research should be done to gather more insight into the way people select and arrange concepts in their workspace. Many user experiments can be performed using the interpretation module by applying different perception theories and their combinations into the work of the agents. In addition, it might be useful to investigate whether the way people write natural language effecting their way in arranging concepts in the workspace, especially in arranging attribute-concepts that explain a main-concept. For example, English is written from left to right, Arabic from right to left, Japanese in columns going from top to bottom and so on.

In sequence with the work of the input interpretation, the overlay operation was able to validate a new input and update the current knowledge, which was based on previous inputs. The approach can be used for integrating a sequence of input messages from a particular user, anaphora analysis based on previous inputs and integrating input messages from multiple users. The use of this operation alone is not sufficient. The current approach combines the overlay operation with the validation based on scripts, especially when the new incoming message contains concepts that are unknown or conflicting with the previous knowledge. The scripts provides mechanisms to infer related concepts in the user message that may be not yet defined (exclusion) or may be defined using other concepts (inclusion). For the overall knowledge integration process, some additional metrics have to be taken into account to consider whether a concept in the incoming message is actually the same concept in the current knowledge, such as time interval, the distance to the reference and the like. Further research could be done on exploring the use of this message integration module in a bigger scale domain.

The visual language-based inputs can be interpreted directly since each icon has a direct link to a concept in the ontology. However, for text input, a language-to-concepts parser is necessary. The proposed parser works well for simple sentences. A compound sentence must be split into two or more simple sentences. Its accuracy highly depends on the coverage concepts in the ontology and the accuracy of its tagger tool. In addition, the results of the parser will be more accurate if the syntax of the inputted text has already been corrected in the input recognition modules.

The proposed text-based emotion analysis is based on a keyword spotting approach by categorizing the emotion into having a positive or negative orientation with an intensity. Besides being simple, this approach is suitable for high uncertainties input recognition with grammatical errors, fragmented input and implicit references. The testing results showed that it can detect the urgency of the user input. As drawbacks, the approach relies on the presence of obvious emotion words in the input and its accuracy depends on the coverage and the accuracy of the affective lexicon database. Moreover, it does not yet cover the direction (or the intention) of the emotion to answer questions, such as "is the user emotionally active toward, directed by or provoked by an object or an event?" and "which object or event is the emotion directed to?". Future work could be done on studying more dialogue corpora and to evaluate the approach with different sentence structures and human users. This may also include emotion analysis from discourse information, such as moods, personality characteristics, anaphoric information and other background contexts.

4. ***How can we specify and produce context-sensitive and user-tailored output?***

The output generation is handled by two modules, an interaction manager (IM) and a fission output, which work in sequence. The test results showed that the developed interaction manager module was able to select a communication act from a list of communication strategy based on the current information about the user and the world. Its work is controlled by a frame-based computational model, which applies rules and strategies for selecting actions that are adaptive to the observed user emotion. Using this model causes the overall module to be rather slot-oriented. With the availability and direct access to the knowledge of the user, the task and the world, the developed fission module was able to generate integrated and dynamic multimodal responses to the user system interaction. The module produces output based on consideration of user context variables, such as location (perspective), available communication modalities and emotional state. This reflects on both the visual and language generation.

Since the user input may be uncertain, the order and timing in which concepts are active and which frames can be filled cannot be determined beforehand. As the IM module only analyzes the input controlled by the frames, the visual generation applies one-to-one concept-to-icon mapping, and the language generation selects the response only controlled by the AIML format, this becomes somewhat easier to manage. The approaches makes decisions and inferences easier to verify afterwards. Another advantage is that even though the frames may be filled in a different order and a different communication act may be selected, the communication strategies and the AIML-based language generation can be designed and specified sequentially in a similar way (or together) with the development of the ontology, the scripts and the icon database. However, as shown by the test results, a major drawback is that all concepts in the ontology, the scripts, the icon database, the communication strategies and the AIML dialogue units have to be designed and specified in advance. For user system interaction that is fully expressive, all possible conditions in the domain of interest have to be converted into ontology, scripts, icons, communication strategies and AIML dialogue units. Developing a large number of communication strategies and AIML dialogue units is necessary to take care of the different structures of the relationships between concepts.

Additionally, another topic for further research could be on mechanisms to select the communication acts using a probabilistic approach. Besides for offering the advantages of coping with uncertainties the input and the user emotion, the next action selection will include the prediction of the shortest frame-filling path over the course of the interaction. This implies an expectation of having a more efficient interaction.

5. ***How can we develop a multimodal interface for handheld devices that accommodates the flexible switching of communication modalities?***

The framework consists of predefined modules that can be connected in an ad hoc fashion. It allows rapid construction of a multimodal, multi-devices and multi-users communication system within a certain domain. Although the current developed framework offers recognition only of text and visual language, it was designed to be able to include other input modalities as long as their recognition results have the same knowledge structure. Each module in the framework can be used in isolation, but also as part of the framework. They can be developed modularly and incrementally depending on the complexity and the domain in focus.

The development of the system built based on the HHI framework has demonstrated that the proposed framework allows a developer to pick and mix input and output modalities, ad-hoc, that are suitable for the requirements and the domain of his/her system. Similar to pick and mix blocks from a construction kit, the developer can construct the system design using the components and modules in the framework to support the chosen input and output modalities.

The user test results also showed that the users took the advantage of the multimodality aspect of the developed interface. Using the current prototype, users can look for icons in the menu based on their image or their `tip-text` or using a search engine based on a keyword; they also can create a visual language message and check its interpretation results in textual form. Future work has to be done on developing the system based on its complete design, which includes allowing the users to create and send a combination of textual and visual language messages and on testing the system using different types of handheld devices in a real mobile setting. In this way, we could analyze how people use and experience different modalities on different types of handheld devices during mobile activities.

6. ***Can the proposed adaptive interface support communication between human users and between the users and computer?***

The answer is *yes, it can.* Although the prototype of the communication AUI on handheld devices has not yet been fully tested for multiple users in a real mobile-crisis setting, the user test results showed that our target users were able to express their messages using the modalities provided. These results also indicated that the developed AUI is able to support both communication between users and communication between users and the underlying system. In line with the development and the testing of the prototype, we have collected underlying theories, structured methodologies and practical guidelines as basis ideas how modules in the HHI framework should be developed. With a predefined problem domain, application goals and selected target users in focus, a developer can follow a similar methodology in modeling and developing the system's knowledge representation to support both communication inter-components in the system and context-awareness requirements. Furthermore, the knowledge regarding technical, social and usability aspects of user requirements in (specifically) AUI and (generally) HHI that has been generated in this research can be dedicated as considerations and guidelines during the design process.

Future work has to be done on the evaluation of the framework and the systems built based on the framework in a more realistic setting. Two different methods of evaluation could be applied. The first method is using simulator software that provides a virtual environment. This approach allows us to fine-tune the complexity and performance modules and such systems. However, a simulation is limited in realism and participants in simulations are likely to behave differently than in real situations. Therefore, as the second method, we propose to evaluate the framework and the systems in a real mobile setting to capture more realistic problems and requirements. The evaluation, with respect to the second direction, will allow us to determine how people use the developed systems and to assess whether they improve the efficiency of a given task. It will also allow us to see how people that experience different emotional and social issues, use the systems, and how the interface of the system aids or interferes with the (traditional) task process.

Two more topics for further research could be on investigating how users deal with multitasking and task interruptions while working on the communication AUI and on exploring ways to allow users dividing, delegating and collaborating their primary tasks in a multi-user and multi-device environment using the communication AUI.

Appendix A
Crisis Scenario

In which the scenario and concepts used in developing knowledge representation of the research demonstrator are presented.

A.1 Setting

Figure A.1 shows the map of the crisis used to build the scenes. The numbers show the movement of the witness.

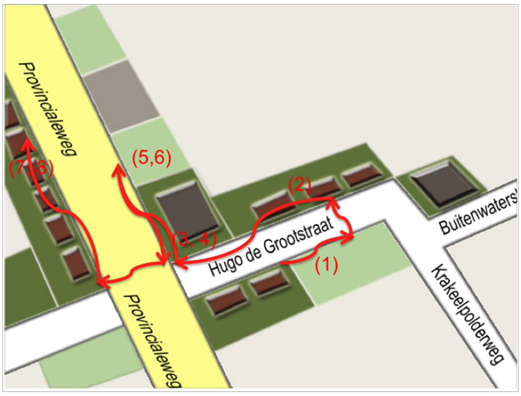


Figure A.1: *The map of the crisis area*

The following is the scenario used during experiments, possible messages and an example of a screenshot. Time legend: mm:ss (time interval).

00:00

It is an ordinary bright morning. People are walking on a path. The traffic is quite busy. People go to work or school. A sudden squeal-sound of tires and steel against the asphalt comes from a distant behind. It is followed by loud banging, crashing, sounding of car horns and screaming. People are running.

Possible messages:

O: An accident near XXX.

O: Traffic jam around XXX.

00:37

From a distance, a long traffic jam has occurred around a intersection. A noise from horns of the inpatient drivers is heard around the area. Suddenly, there is a sound of a scream. People direct their attention to the source of the sound. People seem to be concentrated in a certain area and surrounding a certain object. There is a female person lying on the street.

Possible messages:
O: A female victim at XXX.
C: Policemen come to XXX.
O: A car hits a person.

01:07

Apparently the female person has fainted due to shocked. There is not major problem because soon she is recovered and helped by people around her. However, a few meters from an intersection, a car is wrecked underneath a truck. Other cars tried to avoid

collision with each other. Some cars had to break suddenly. This causes traffic jams on both sides of the road. Policemen come in cars and on motorcycles. Crowds of people are witnessing the event. Everyone stays back, nobody dares to come close.
Possible messages:
O: A collision between a car and a truck.
O: Policemen are here.
O: People around XXX.
C: Ambulances come to XXX.

01:40

The policemen immediately block off the road. Some of them clear the road and get the traffic moving from the other side of the road and behind the scene. An injured victim is found. He is a passenger of the wrecked car. A policeman secures the truck driver to aside. An ambulance comes to scene. A paramedic helps the injured victims. He takes the victim to the hospital in his ambulance. The policemen continue their work to clear the road.

Possible messages:
P: The road (XXX) is blocked off.
O: Some Policemen clear the road.
P: We cannot move the car and the truck aside.
O: There is one male victim.
O: The ambulance is here.
A: We found one injured victim. One male.
P: One injured victim is secured.
A: We go to the hospital.
O: The paramedic takes the victim to his ambulance.
O: The paramedic drives his ambulance to the hospital.

02: 25

Some Policemen run to see another car behind the truck. This car is flipped over. The driver is still inside unconscious. At the side of the road, there is a gas station. The Policemen open the car and help the driver out. People come forward curious to know more, but a policeman asks them to leave the accident area. They are pushed away more to the side.

Possible messages:
O: I see another car flipped over behind the truck.
P: We found another injury. One male. He is unconscious. He is still inside the car. The gas is leaking. It is dangerous. We need firefighters.
P: We cannot move this car too.
A: Another injury. One male.
A: We bring the injuries to the hospital.
C: Firefighters come to XXX.

03:01

An ambulance hass appeared behind a truck and the flipped car. A traffic policeman is

still busy clearing the road. People are pushed away more to the side. Both sides of the road are empty, except those wrecked vehicles. Two Policemen are busy with helping the driver inside the flipped car. They try to open the car and help the driver out. There is smoke and fire ignition from the car. Finally, the driver is out of the car. A paramedic comes forward to help the policemen. Possible messages:
O: An ambulance is here.
O: The victim is still inside the car.
O: Two policemen help the injuries.
O: The driver is taken by a policeman.
O: The driver is out of the car.
O: Smoke out of the car.
O: The car is on fire.
P: The situation is dangerous.
C: The area of XXX is not safe. Please stay alert.

03: 38
When the Policemen and the driver are about few meters away from the car, suddenly it explodes. A policeman and the driver are injured badly. The paramedic brings them to the hospital right away.
Possible messages:
O: There is an explosion.
P: The car is exploded.
P: The area is very dangerous.
P: One male injury. One officer is down.
C: More Policemen and ambulances come to XXX.
C: Firefighters on their way to XXX.
C: Please stay away from XXX. This area is very dangerous.
A: Two injuries. One of them is an officer. I take them to the hospital.
C: Firefighters come to XXX.

04:11

The explosion causes fire in the area. The car and the truck are smashed and burnt. At the side of the road, the gas station is also half burnt. A Fire-truck is coming. The firemen are busy putting out the fire around the area. The area covers with smoke. After a while, the fire seems under control. Suddenly, the firemen move away from the area leaving their equipment. Soon, there is a bigger explosion coming from the gas station.

Possible messages:

O: The truck and the car are burnt.

P: Fire on the gas station.

P: The gas station contains flammable items.

O: I see smoke.

F: Fire covers a large area of the gas station.

F: The area is too dangerous.

F: Another explosion may occur.

C: The area XXX (larger) is blocked for public access.

A.2 The Taxonomy of Recognized Concepts

The following are concepts found in the scenario.

```
Role
    Event
         PhysicalObject
              WorldObject
                    DynamicObject
                          CrisisEvent
                                    Accident
                                    Fire
                                    Explosion
                                    Smoke
                                    TrafficJam
                                    GasLeak
                          Actor
                                    Person (Unknown gender)
                                    Male
                                    Female
                                    Victim (Injured person)
                                    People (Multiple person)
                                    Policeman
                                    Firefighter
                                    Paramedic
                                    Passenger
                          Transportation
                                    Car
                                    Truck
                                    Motorcycle
                                    PoliceCar
                                    FireTruck
                                    Ambulance
                      StaticObject
                            Building
                                    House
                                    Shop
                                    GasStation
                                    Hospital
                           Street
                                   Intersection
         Process
               GeneralProcess
                     Select
                MentalProcess
                     SensorAct
                         See
                         Hear
                PhysicalProcess
                    RescueEvent
                          RegulateTraffic
                          Evacuate
                          PutFire
                     TransferAct
                          Give
                          Take
                      MovementAct
                          Come
                          Move
                          Go
                          Drive
                SocialProcess
```

```
                       CommunicationAct
                          State
                          Request
                          Ask
                          Acknowledge
                          Confirm
                          Highlight
                          Update
                          Affirm
                          Negate
     AbstractEvent
          AbstractObject
                 AbstractNumber
                       Many
                       Few
                       Some
                 AbstractRepresentationalObject
                       Causality
                             HeartAttack
                             Burnt
                             BrokenNeck
                       Attribute
                             Address
                             Color
                                    Black ... White
                             Gender
                             State
                                    CapacityState
                                    DynamicSpatialState
                                    HazardousLevel
                                    IsTrapped
                                    LivingCondition
                                    UrgencyLevel
                                    SubstrateCondition
                       Measurement
                            Size
                                    Small
                                    Big
          Weight
                                    Heavy
                                    Light
                       Relation
                             SpatialRelation
                                   TwoPointRelation
                                         Inside
                                         Outside
          AbstractProcess
                 Negation
Type
         AbstractEventType
                 AbstractObjectType
                            Number (0..9)
```

Bibliography

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Proc. of HUC '99*, volume 1707 of *Lecture Notes In Computer Science*, pages 304–307, London, UK. Springer-Verlag.

Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, **16**, 3–9.

Alexandersson, J. & Becker, T. (2003). The formal foundations underlying overlay. In *Proc. of IWCS-5 2003*, pages 22–36, Tilburg, the Netherlands.

Alexandersson, J., Becker, T., & Pfleger, N. (2006). Overlay: The basic operation for discourse processing. In W. Wahlster, editor, *SmartKom - Foundations of Multimodal Dialogue Systems*, Cognitive Technologies, pages 255–267. Springer, Berlin, Heidelberg.

Altmann, E. M. & Gray, W. D. (2000). An integrated mode of set shifting and maintenance. In *Proc. of ICCM 2000*, pages 17–24, the Netherlands. Universal Press.

Alvarez-Cortes, V., Zayas-Perez, B. E., Zarate-Silva, V. H., & Ramirez Uresti, J. A. (2007). Current trends in adaptive user interfaces: Challenges and applications. In *Proc. of CERMA '07*, pages 312–317, Washington, DC, USA. IEEE Computer Society.

Amsbary, P. (2007). Museum recorder for the high museum, http://lcc.gatech.edu/ pamsbary/.

Anson, D. K., Moist, P., Przywara, M., Wells, H., Saylor, H., & Maxime, H. (2006). The effects of word completion and word prediction on typing rates using on-screen keyboards. *Assistive Technology*, **18**, 146–154.

Apple-Computer, Inc. (1992). *Macintosh Human Interface Guidelines*. Addison-Wesley Publishing Company, USA.

Arnheim, R. (1969). *Visual Thinking*. University of California Press, Berkeley, Los Angeles, London.

AvantGo (2003). Research summary: 2003 mobile lifestyle survey.

Barrow, H. & Baker, B. (1982). Minspeak: A semantic compaction system that makes self expression easier for communicatively disabled individuals. *Byte*, **7**(9), 186–202.

Barsalou, L. W. (1992). *Cognitive Psychology*. Erlbaum, Hillsdale, NJ.

Bavelier, D., Corina, D. P., & Neville, H. J. (1998). Brain and language: a perspective from sign language. *Neuron*, **21**, 275–278.

Beardon, C. (1992). CD-Icon: an iconic language-based on conceptual dependency. *Intelligent Tutoring Media*, **3**(4), 111–116.

Beaudouin-Lafon, M. & Conversy, S. (1996). Auditory illusions for audio feedback. In M. Tauber, editor, *CHI '96: Conference companion on Human factors in computing systems*, pages 299–300, New York, USA. ACM Press.

Bell, G. & Gray, J. N. (1997). The revolution yet to happen. *Beyond Calculation: the Next Fifty Years*, pages 5–32.

Bellotti, V. & Bly, S. (1996). Walking away from the desktop computer: distributed collaboration and mobility in a product design team. In *Proc. of CSCW '96*, pages 209–218, New York, USA. ACM Press.

Benjamins, T. (2006). *MACSIM: Multi Agent Crisis Simulator, Interpreter, and Monitor*. Master's thesis, Delft University of Technology, the Netherlands.

Berger, J. (1972). *Ways of Seeing*. Britain Broadcasting Corp., London, UK.

Bernsen, N. O. (1997). Defining a taxonomy of output modalities from an HCI perspective. *Computer Standard Interfaces*, **18**(6-7), 537–553.

Biem, A. (2006). Minimum classification error training for online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(7), 1041–1051.

Billsus, D., Brunk, C. A., Evans, C., Gladish, B., & Pazzani, M. (2002). Adaptive interfaces for ubiquitous web access. *Communication ACM*, **45**(5), 34–38.

Bliss, C. K. (1984). *The Blissymbols Picture Book*. Semantography Press, Sidney, Australia.

BNC (2007). A British National Corpus: Unigrams and bigrams, Retrieved on January 5, http://natcorp.ox.ac.uk.

Bohan, M., Phipps, C. A., Chaparro, A., & Halcomb, C. G. (1999). A psychophysical comparison of two stylus-driven soft keyboards. In I. S. MacKenzie and J. Stewart, editors, *Graphics Interface*, pages 92–97, Kingston, Ontario, CA. Canadian HCC Society.

Boissiére, P. & Dours, D. (1996). Vitipi: Versatile Interpretation of Text Input by Persons with Impairments. In *Proc. of ICCHP 1996*, pages 165–172, Linz, Austria.

Boren, T. & Ramey, J. (2000). Thinking aloud: reconciling theory and practice. *IEEE Transactions on Professional Communication*, **43**(3), 261–278.

Bosma, W. & Andre, E. (2004). Exploiting emotions to disambiguate dialogue acts. In *Proc. of IUI 2004*, pages 85–92, Portugal.

Bousquet-Vernhettes, C., Privat, R., & Vigouroux, N. (2003). Error handling in spoken dialogue systems: Toward corrective dialogue. In *Proc. of ISCA '03*, pages 95–100, San Diego, Ca, USA. IEEE Computer Society.

Brar, G. S., Biswas, S., Kundu, S., Mukhopadhyay, A., Worah, P., & Basu, A. (2004). OaSis: An application specific operating system for an embedded environment. In *Proc. of VLSID '04*, pages 776–779, Washington, DC, USA. IEEE Computer Society.

Browne, D., Norman, M., & Riches, D. (1990). Why build adaptive systems? In D. Browne, P. Totterdell, and M. Norman, editors, *Adaptive User Interfaces*, pages 15–57. Academic Press, London, UK.

Bui, T. H., Poel, M., Nijholt, A., & Zwiers, J. (2007). A tractable DDN-POMDP approach to affective dialogue modeling for general probabilistic frame-based dialogue systems. In D. Traum, J. Alexandersson, A. Jonsson, and I. Zukerman, editors, *Proc. of IJCAI '07*, pages 34–37, Hiderabad, India.

Card, S. K., Newell, A., & Moran, T. P. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA.

Carmel, E., Crawford, S., & Chen, H. (1992). Browsing in hypertext : a cognitive study. *IEEE Transactions on Systems, Man, and Cybernetics*, **22**(5), 865–884.

Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press, New York, USA.

Cassell, J. (2000). *Embodied Conversational Agents*. The MIT Press, Cambridge, MA.

Cassell, J., Vilhjálmsson, H. H., & Bickmore, T. (2001). BEAT: the Behavior Expression Animation Toolkit. In *Proc. of SIGGRAPH '01*, pages 477–486, New York, NY, USA. ACM Press.

Cassell, J., Stocky, T., Bickmore, T., Gao, Y., Nakano, Y., Ryokai, K., Tversky, D., Vaucelle, C., & Vilhjlmsson, H. (2002). MACK: Media lab Autonomous Conversational Kiosk. In *Proc. of Imagina 02: Intelligent Autonomous Agents*, 'Monte Carlo, USA.

Chandler, D. (2001). *Semiotics: The Basics*. Routledge, New York, USA.

Chang, S. K., Polese, G., Orefice, S., & Tucci, M. (1994). A methodology and interactive environment for iconic language design. *International Journal Human Computer Studies*, **41**(5), 683–716.

Chen, G. & Kotz, D. (2000). A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chin, G. J., Stephan, E. G., Gracio, D. K., Kuchar, O. A., Whitney, P. D., & Schuchardt, K. L. (2006). Developing concept-based user interfaces for scientific computing. *Computer*, **39**(9), 26–34.

Clore, G. L. (1992). Cognitive phenomenology: Feelings and the construction of judgment. In L. Martin and A. Tesser, editors, *The Construction of Social Judgments*, pages 217–245. Lawrence Erlbaum Associates.

Cohn, N. (2003). Visual syntactic structures, towards a generative grammar of visual language. Technical report, Emaki.

Comerford, L., Frank, D., Gopalakrishnan, P., Gopinath, R., & Sedivy, J. (2001). The IBM Personal Speech Assistant. In *Proc. of IEEE ICASSP 2001*, pages 1–4, Salt Lake City, USA. IEEE Computer Society.

Condon, W. S. & Osgton, W. D. (1971). Speech and body motion synchrony of the speaker-hearer. In D. Horton and J. Jenkins, editors, *The Perception of Language*, pages 150–184. Academic Press, New York, USA.

Corballis, M. C. (2002). Did language evolve from manual gestures? In A. Wray, editor, *The Transition to Language*, chapter 8. Oxford University Press, Oxford, UK.

Corrada-Emmanuel, A. (2007). $n.d.$. enron e-mail dataset research. Retrieved in January 5, 2007, http://ciir.cs.umass.edu/c̃orrada/enron/.

Coutaz, J., Nigay, L., & Salber, D. (1993). Taxonomic issues for multimodal and multimedia interactive systems. In *Proc. of the ERCIM Workshop on Multimodal HCI*, pages 3–11, Nancy, France.

Crawford, C. (2008). Deikto: an application of the weak sapir-whorf hypothesis. In *Creating '08: Proceedings of the hypertext 2008 workshop on Creating out of the machine: hypertext, hypermedia, and web artists explore the craft*, pages 1–4, New York, NY, USA. ACM.

Cutrell, E. B., Czerwinski, M., & Horvitz, E. (2000). Effects of instant messaging interruptions on computing tasks. In *Proc. of CHI 2000*, pages 99–100, New York, NY, USA. ACM.

Damasio, A. (1994). *Descartes' Error - Emotion, Reason, and the Human Brain*. Grosset/Putnam, New York.

Das, S. R. & Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, **53**(9), 1375–1388.

Demers, A. J. (1994). Research issues in ubiquitous computing. In *Proc. of PODC '94*, pages 2–8, New York, USA. ACM Press.

Desmet, P. (2002). *Designing Emotion*. Ph.D. thesis, Delft University of Technology, the Netherlands.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human Computer Interaction*, **16**(2), 97–166.

Dix, A. J., Finlay, J., & Abowd, G. D. (2004). *Human-Computer Interaction*. Pearson Prentice-Hall, Harlow, UK, 3rd edition.

Dor, R. (2005). *The Ear's Mind, a Computer Model of the Fundamental Mechanisms of the Perception of Sound.* Master's thesis, Delft University of Technology.

Doran, C., Hockey, B. A., Sarkar, A., Ann, B., Sarkar, H. A., Srinivas, B., & Xia, F. (2000). Evolution of the XTAG system. In *Proc. of TAG+5*, pages 371–403. CSLI publications.

Dusan, S., Gadbois, G., & Flanagan, J. (2003). Multimodal interaction on pda's integrating speech and pen inputs. In *Proc. of Eurospeech '03*, pages 2225–2228, Geneva, Switzerland.

Dymon, U. J. (2003). An analysis of emergency map symbology. *International Journal of Emergency Management*, **1**(3), 227–237.

Ekman, P. (1979). About brows, emotional and conversational signals. In M. von Cranach, K. Foppa, W. Lepenies, and D. Ploog, editors, *Human Ethology, Claims and Limits of a New Discipline, Contributions to the Colloquium*, pages 169–248. Cambridge University Press.

Ekman, P. (1999). Basic emotions. In T. Dalgleish and M. Power, editors, *Handbook of Cognition and Emotion*, pages 45–60. John Wiley and Sons Ltd., Sussex, UK.

Ekman, P. (2001). *Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage.* W. W. Norton & Co., New York, USA & London, UK, 2 rev sub edition.

Elliot, C. D. (1992). *The Affective Reasoner: a Process Model of Emotions in a Multi-Agent System.* Ph.D. thesis, Northwestern University, Evanston, Illionis.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **37**(33), 32–64.

Farberow, N. & Frederick, C. (1978). Training manual for human service workers in major disasters. Technical report, Maryland: National Institute of Mental Health, Rockville.

Fazly, A. & Hirst, G. (2003). Testing the efficacy of part-of-speech information in word completion. In *Proc. of EACL 2003*, pages 9–16, Budapest, Hungary.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication).* The MIT Press, Cambridge, MA.

Fillmore, C. (1968). The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, New York.

Fitrianie, S. (2004). *An Icon-based Communication Interface on a PDA.* Master's thesis, Post-Graduate Programme of User System Interaction, Eindhoven University of Technology.

Fitrianie, S. & Rothkrantz, L. J. M. (2007a). An adaptive keyboard with personalized language-based features. In *Proc. of TSD 2007*, volume 4629 of *Lecture Notes in Computer Science*, pages 131–138, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2007b). An automated text-based synthetic face with emotions for web lectures. *Communication and Cognition - Artificial Intelligence*, **23**(1–4), 89–98.

Fitrianie, S. & Rothkrantz, L. J. M. (2008a). An automated online crisis dispatcher. *International Journal of Emergency Management*, **5**(1–2), 123–144.

Fitrianie, S. & Rothkrantz, L. J. M. (2008b). The generation of emotional expressions for a text-based dialogue agent. In *Proc. of TSD 2008*, volume 5246 of *Lecture Notes in Computer Science*, pages 569–576, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2009a). Computed ontology-based situation awareness of multi-user observations. In J. Landgren and S. Jul, editors, *Proc. of ISCRAM 2009*, Gothenburg, Sweden.

Fitrianie, S. & Rothkrantz, L. J. M. (2009b). Database for a visual language-based application. In *Proc. of Euromedia 2009*, pages 10–17, Bruges, Belgium. Eurosis.

Fitrianie, S., Wiggers, P., & Rothkrantz, L. J. M. (2003). A multi-modal eliza using natural language processing and emotion recognition. In *Proc. of TSD 2003*, volume 2807 of *Lecture Notes in Computer Science*, pages 394–399, Berlin, Heidelberg. Springer/Verlag.

Fitrianie, S., Datcu, D., & Rothkrantz, L. J. M. (2006). Constructing knowledge of the world in crisis situations using visual language. In *Proc. of IEEE SMC 2006*, pages 121–126. IEEE Computer Society.

Fitrianie, S., Poppe, R., Bui, T. H., Chiţu, A. G., Datcu, D., Dor, R., Hofs, D. H. W., Wiggers, P., Willems, D. J. M., Poel, M., Rothkrantz, L. J. M., Vuurpijl, L. G., & Zwiers, J. (2007). A multimodal human-computer interaction framework for research into crisis management. In v. d. B. Walle, P. Burghardt, and K. Nieuwenhuis, editors, *Proc. of ISCRAM 2007*, pages 149–158, Delft, the Netherlands.

Fitrianie, S., Tatomir, I., & Rothkrantz, L. J. M. (2008a). A context aware and user tailored multimodal information generation in a multimodal HCI framework. In *Proc. of Euromedia 2008*, pages 95–103, Porto, Portugal. Eurosis.

Fitrianie, S., Yang, Z., & Rothkrantz, L. J. M. (2008b). Developing concept-based user interface using icons for reporting observations. In F. Friedrich and B. van de Walle, editors, *Proc. of ISCRAM 2008*, pages 394–403, Washington, DC, USA.

Fitrianie, S., Yang, Z., Datcu, D., Chiţu, A. G., & Rothkrantz, L. J. M. (2010). Context-aware multimodal human-computer interaction. In R. Babuska and F. C. A. Groen, editors, *Interactive Collaborative Information Systems*, volume 281 of *Studies in Computational Intelligence*, pages 237–272. Springer.

Fitts, P. & Posner, M. (1967). *Human Performance.* Wadsworth, Wokingham, UK.

Forgy, C. L. (1990). Rete: a fast algorithm for the many pattern/many object pattern match problem. *Expert Systems: a Software Methodology for Modern Applications*, pages 324–341.

Foster, M. E., White, M., Setzer, A., & Catizone, R. (2005). Multimodal generation in the COMIC dialogue system. In *Proc. of ACL '05*, pages 45–48, Morristown, NJ, USA. Association for Computational Linguistics.

Frokjaer, E., Hertzum, M., & Hornbaek, K. (2000). Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proc. of SIG CHI '00*, pages 345–352, New York, USA. ACM Press.

Frutiger, A. (1989). *Sign and Symbols, Their Design and Meaning*. van Nostrand Reinholt, New York.

Garay-Vitoria, N. & Abascal, J. (2004). A comparison of prediction techniques to enhance the communication rate. In C. Stary and C. Stephanidis, editors, *User Interfaces for All*, volume 3196 of *Lecture Notes in Computer Science*, pages 400–417. Springer.

Ghose, S. & Dou, W. (1998). Interactive functions and their impacts on the appeal of internet presences sites. *Journal of Advertising Research*, **38**, 29–43.

Goldberg, D. & Goodisman, A. (1991). Stylus user interfaces for manipulating text. In *Proc. of UIST '91*, pages 127–135, New York, USA. ACM Press.

Goonetilleke, R. S., Shih, H. M., On, H. K., & Fritsch, J. (2001). Effects of training and representational characteristics in icon design. *International Journal of Human-Computer Studies*, **55**(5), 741–760.

Gopher, D. & Donchin, E. (1986). Workload: An examination of the concept. In K. R. Boff, L. Kaufman, and J. P. Thomas, editors, *Handbook of Perception and Human Performance*, volume II: Cognitive Processes and Performance, pages 270–319. Wiley, New York, USA.

Gorlenko, L. & Merrick, R. (2003). No wires attached: Usability challenges in the connected mobile world. *IBM System Journal*, **42**(4), 639–651.

Gould, J. D. (1988). How to design usable systems. In M. Helander, editor, *Handbook of Human-Computer Interaction*, pages 757–789. North-Holland, Amsterdam, NL.

Guenthner, F., Kr§ger-Thielmann, K., Pasero, R., & Sabatier, P. (1993). Communication aids for handicapped persons. In *Proc. of ECART 1993*, pages 1.4.1 – 1.4.3, Stockholm, Sweden.

Gurevych, I., Merten, S., & Porzel, R. (2003). Automatic creation of interface specifications from ontologies. In *Proc. of HLT-NAACL SEALTS '03*, pages 59–66, Morristown, NJ, USA. Association for Computational Linguistics.

Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granstrm, B., House, D., & Wirn, M. (2000). AdApt - a multimodal conversational dialogue system in an apartment domain. In *Proc. of ICSLP 2000*, volume 2, pages 134–137, Beijing, China.

Hassanein, K. & Head, M. (2003). Ubiquitous usability: Exploring mobile interfaces within the context of a theoretical model. In *Proc. of CAiSE 2003*, pages 180–194, Velden, Austria.

Hatzivassiloglou, V. & McKeown, K. (1997). Predicting the semantic orientation of adjectives. In *Proc. of ACL 1997*, pages 174–181, Morristown, NJ, USA. Association for Computational Linguistics.

Heidorn, G. (2000). Intelligent writing assistant. In R. Dale, H. Moisl, and H. Somers, editors, *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, pages 305–327. Marcel Dekker, New York, USA.

Heinich, R., Molenda, M., Russell, J. D., & Smaldino, S. E. (1999). *Instructional Media and Technologies for Learning*. Prentice-Hall, Upper Saddle River, NJ, 6th edition.

Henricksen, K., Indulska, J., & Rakotonirainy, A. (2002). Modeling context information in pervasive computing systems. In *Proc. of Pervasive '02*, volume 2414 of *Lecture Notes in Computer Science*, pages 167–180, London, UK. Springer-Verlag.

Hofstadter, D. R. (1996). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, Inc., New York, USA.

Holcomb, R. & Tharp, A. L. (1991). What users say about software usability. *International Journal of Human-Computer Interaction*, **3**(1), 49–78.

Horton, W. K. (1994). *The ICON Book: Visual Symbols for Computer Systems and Documentation*. John Wiley & Sons, Inc., New York, NY, USA.

Housz, T. I. (1994). The elephant's memory, http://www.khm.de/timot.

HSWG (2003). Symbology reference. homeland security working group, http://www.fgdc.gov/hswg/index.html.

Hu, M. & Liu, B. (2004). Mining and summarizing customer reviews. In *Proc. of ACM SIGKDD '04*, pages 168–177, New York, USA. ACM Press.

Hudlicka, E. (2003). To feel or not to feel: The role of affect in human-computer interaction. *International Journal Human-Computer Studies*, **59**(1–2), 1–32.

IBM.com (1999). IBM e-business case studies: Bullhead city police department, http://www.ibm.com/e-business/doc/content/casestudy/47405.html.

ISO/IEC (2000). Information technology, user system interfaces and symbols, icon symbols and functions. 1st edition, iso/iec 11581-1:2000(e) to iso/iec 11581-6:2000(e). Technical report, International Standard.

Isokoski, P. (2004). *Manual Text Entry: Experiments, Models, and Systems*. Ph.D. thesis, Department of Computer Sciences, University of Tampere, Finland.

Jaimes, A. & Sebe, N. (2007). Multimodal human-computer interaction: A survey. *Computer Visual Image Understanding*, **108**(1–2), 116–134.

Jameson, A. (2002). Usability issues and methods for mobile multimodal systems. In *Proc. of ISCA 2002 on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany.

Johanson, B., Fox, A., & Winograd, T. (2002). The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, **1**(2), 67–74.

Johnston, M. & Bangalore, S. (2000). Finite-state multimodal parsing and understanding. In *Proc. of ACL 2000*, pages 369–375, Morristown, NJ, USA. Association for Computational Linguistics.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., & Maloor, P. (2001). MATCH: an architecture for multimodal dialogue systems. In *Proc. of ACL '02*, pages 376–383, Morristown, NJ, USA. Association for Computational Linguistics.

Jones, M. & Marsden, G. (2006). *Mobile Interaction Design*. John Wiley & Sons, New York, USA.

Joshi, A. K. (2001). The XTAG project at Penn. In *Proc. of IWPT-2001*, Beijing, China.

Joshi, A. K. & Schabes, Y. (1997). Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, Berlin, New York.

Joshi, A. K. & Vijay-Shanker, K. (1999). Compositional semantics for Lexicalized Tree-Adjoining Grammars: How much underspecification is necessary? In *Proc. of Computational Semantics 1999*, pages 131–145, the Netherlands.

Kamps, J. & Marx, M. (2002). Words with attitude. In *Proc. of Global WordNet CIIL'02*, pages 332–341, Mysore, India.

Karat, C.-M., Halverson, C., Horn, D., & Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proc. of SIGCHI '99*, pages 568–575, New York, NY, USA. ACM Press.

Kärkkäinen, L. & Laarni, J. (2002). Designing for small display screens. In *Proc. of NordiCHI '02*, pages 227–230, New York, USA. ACM Press.

Karlson, A., Bederson, B., & Contreras-Vidal, J. (2006). Understanding one-handed use of handheld devices. In L. Jo, editor, *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, pages 86–100. Idea Group.

Keizer, S. (2003). *Reasoning under uncertainty in Natural Language Dialogue using Bayesian Networks*. Ph.D. thesis, Twente University, the Netherlands.

Kendon, A. (1980). Gesture and speech: Two aspects of the process of the utterance. *The Relation Between Verbal and Non-Verbal Communication*, pages 207–227.

Keyes, E., Sykes, D., & Lewis, E. (1989). Technology + design + research = information design. In *Text, Context, and Hypertext*, pages 251–264. The MIT Press, Cambridge, MA.

Kim, H., Kim, J., Lee, Y., Chae, M., & Choi, Y. (2002). An empirical study of the use contexts and usability problems in mobile internet. In *Proc. of HICSS '02*, pages 1767–1776, Washington, DC, USA. IEEE Computer Society.

Kim, S.-M. & Hovy, E. (2005). Automatic detection of opinion bearing words and sentences. In *Proc. of IJCNLP 2005 Companion Volume*.

Kim, S.-M. & Hovy, E. (2006). Identifying and analyzing judgment opinions. In *Proc. of NAACL HLT 2006*, pages 200–207, Morristown, NJ, USA. Association for Computational Linguistics.

Kirakowsky, J. (1996). The software usability measurement inventory: Background and usage. In B. A. W. P. W. Jordan, B. Thomas and I. L. McClelland, editors, *Usability Evaluation in Industry*, pages 169–178. London, Taylor and Francis, UK.

Kjeldskov, J. & Kolbe, N. (2002). Interaction design for handheld computers. In *Proc. of AP-CHI'02*, pages 433–446, China. Science Press.

Klapwijk, P. (2005). *TICS: A Topology based Infrastructure for Crisis Situations.* Master's thesis, Delft University of Technology.

Kosslyn, S. M. (1994). *Image and Brain - The Resolution of the Imagery Debate.* MIT Press, Cambridge, MA.

Kranstedt, A., Kopp, S., & Wachsmuth, I. (2002). MURML: A Multimodal Utterance Representation Markup Language for conversational agents. In *Proc. of AAMAS '02, Workshop on ÒEmbodied conversational agents Ð LetÕs specify and evaluate them!Ó*, Bologna, Italy.

Krauss, R. M. & Fussell, S. R. (1991). Perspective-taking in communication: Representations of others' knowledge in reference. *Social Cognition*, pages 2–24.

Krauss, R. M. & Fussell, S. R. (1996). Social psychological models of interpersonal communication. In E. T. Higgins and A. Kruglanski, editors, *Social psychology: A Handbook of Basic Principles*, pages 655–701. Guilford, New York.

Kristoffersen, S. & Ljungberg, F. (1999). "making place" to make IT work: empirical explorations of HCI for mobile CSCW. In *Proc. of SIG GROUP '99*, pages 276–285, New York, USA. ACM Press.

Kühn, M. & Garbe, J. (2001). Predictive and highly ambiguous typing for a severely speech and motion impaired user. In *Proc. of UAHCI 2001*, pages 933–936, New Orleans, LA, USA.

Kurtenbach, G. & Buxton, W. (1993). The limits of expert performance using hierarchic marking menus. In *Proc. of INTERACT and CHI '93*, pages 482–487, New York, USA. ACM Press.

LaLomia, M. (1994). User acceptance of handwritten recognition accuracy. In *Proc. of SIG CHI '84 (Conference Companion)*, pages 107–108, New York, USA. ACM Press.

Langendorf, D. J. (1988). *Textware Solution's Fitaly Keyboard V1.0 Easing the Burden of Keyboard Input.* WinCELair Review.

Leemans, N. E. M. P. (2001). *VIL: A Visual Inter Lingua.* Ph.D. thesis, Worcester Polytechnic Institute, USA.

Lesher, G. & Rinkus, G. (2001). Domain-specific word prediction for augmentative communication. In *Proc. of the RESNA 2001*, pages 61–63, Reno, Nevada, USA.

Lester, P. M. (2006). *Visual Communication: Images with Messages.* Wadsworth/Thompson, Belmont, CA, 4 edition.

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation.* MIT Press, Cambridge, MA.

Littlejohn, S. W. (2001). *Theories of Human Communication (with InfoTrac) (Wadsworth Series in Speech Communication).* Wadsworth Publishing.

Liu, H., Lieberman, H., & Selker, T. (2003). A model of textual affect sensing using real-world knowledge. In *Proc. of IUI '03*, pages 125–132, New York, USA. ACM Press.

Love, S. (2005). *Understanding Mobile Human-Computer Interaction (Information Systems Series (ISS)).* Butterworth-Heinemann, Newton, MA, USA.

MacKenzie, I. S. & Chang, L. (1999). A performance comparison of two handwriting recognizers. *Interacting with Computers*, **11**(3), 283–297.

Mackenzie, S. I. & Soukoreff, W. R. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, **17**(2 & 3), 147–198.

Maier, A., Haderlein, T., & Nöth, E. (2006). Environmental adaptation with a small data set of the target domain. In P. Sojka, I. Kopecek, and K. Pala, editors, *Proc. of TSD 2006*, number 4188 in Lecture Notes in Artificial Intelligence, pages 431–437, Berlin, Heidelberg, New York. Springer.

Mankoff, J. & Abowd, G. D. (1998). Cirrin: a word-level unistroke keyboard for pen input. In *Proc. of UIST '98*, pages 213–214, New York, USA. ACM Press.

Matiasek, J. & Baroni, M. (2003). Exploiting long distance collocational relations in predictive typing. In *Proc. of EACLÕ03 in Language Modelling for Text Entry Methods*, pages 1–8, Budapest, Hungary.

Maybury, M. T. & Wahlster, W. (1998). Intelligent user interfaces: an introduction. *Readings in Intelligent User Interfaces*, pages 1–13.

McTear, M. F. (2000). Intelligent interface technology: from theory to reality? *Interacting with Computers*, **12**(4), 323–336.

McTear, M. F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Computer Survey*, **34**(1), 90–169.

Mehrotra, S., Butts, C., Kalashnikov, D. V., Venkatasubramanian, N., Altintas, K., Hariharan, R., Lee, H., Ma, Y., Myers, A., Wickramasuriya, J., Eguchi, R., & Huyck, C. (2004). CAMAS: A citizen awareness system for crisis mitigation. In *Proc. of ACM SIGMOD 2004*, pages 955–956, New York, NY, USA. ACM Press.

MESA-Project (2001). Mobile broadband for Emergency and Safety Applications project, http://www.projectmesa.org.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the Structure of Behavior.* Holt, Rinehart and Winston, New York, USA.

Mitchell, M. (1993). *Analogy-Making as Perception: a Computer Model.* MIT Press, Cambridge, MA, USA.

Moore, L. K. (2006). Crs report for congress: Public safety communication policy. Technical report, Confessional Research Service, the Library of Congress.

Mulder, M., Nijholt, A., op den Akker, H., den Uyl, M., & Terpstra, P. (2004). A lexical, grammatical implementation of affect. In *Proc. of TSD 2004*, volume 3206 of *Lecture Notes in Computer Science*, pages 171–177, Heidelberg. Springer Verlag.

Mullen, A. & Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In D. Lin and D. Wu, editors, *Proc. of EMNLP 2004*, pages 412–418, Barcelona, Spain. Association for Computational Linguistics.

Nagao, K. & Takeuchi, A. (1994). Speech dialogue with facial displays: Multimodal human-computer conversation. In *Proc. of ACL 1994*, pages 102–109. Morgan Kaufmann.

Nagata, S. F. & van Oostendorp, H. (2003). Multitasking in a mobile context. In . E. O. P. Gray, H. Johnson, editor, *Proc. of HCI 2003 Designing for Society*, pages 145 – 146, Crete, Greece. Bristol: Research Press International.

Nakatsu, R. (1998). Social, psychological and artistic aspects of the human interface. In *Community Computing and Support Systems, Social Interaction in Networked Communities*, volume 1519 of *Lecture Notes in Computer Science*, pages 94–107, London, UK. Springer-Verlag.

Nasukawa, T. & Yi, J. (2003). Sentiment analysis: capturing favorability using natural language processing. In *Proc. of K-CAP '03*, pages 70–77, New York, USA. ACM Press.

Nelson, L., Bly, S., & Sokoler, T. (2001). Quiet calls: talking silently on mobile phones. In *Proc. of SIG CHI '01*, pages 174–181, New York, USA. ACM Press.

Nielsen, J. (1995). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Nock, H. J., Iyengar, G., & Neti, C. (2004). Multimodal processing by finding common cause. *Communication ACM*, **47**(1), 51–56.

Norman, K. L. (1991). *The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface*. Greenwood Publishing Group Inc., Westport, CT, USA.

Novak, J. D. & Canas, A. J. (2008). Ihmc cmaptools 2006-01 rev 2008-01: The theory underlying concept maps and how to construct and use them. Technical report, Florida Institute for Human and Machine Cognition.

Ogden, C. K. (1930). *Basic English - A General Introduction with Rules and Grammar*. Treber, London, UK.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press, New York, USA.

Otten, J., van Heijningen, B., & Lafortune, J. F. (2004). The virtual crisis management center - an ict implementation to canalize information! In B. Carle and B. V. de Walle, editors, *Proc. of ISCRAM 2004*, pages 21–26, Brussels, Belgium.

Oviatt, S. & van Gent, R. (1996). Error resolution during multimodal human-computer interaction. In *Proc. of ICSLP 1996*, volume 1, pages 204–207, Philadelphia, Pennsylvania.

Oviatt, S. L. (2000). Multimodal system processing in mobile environments. In *Proc. of UIST 2000*, pages 21–30, New York, USA. ACM Press.

Oviatt, S. L. (2001). Designing robust multimodal systems for diverse users and environments. In C. Stephanidis, editor, *Proc. of HCI International 2001*, pages 407–411, New Orleans, USA. Lawrence Erlbaum.

Oviatt, S. L., Darrell, T., & Flickner, M. (2004). Multimodal interfaces that flex, adapt, and persist. *Communication ACM*, **47**(1), 30–33.

Pang, B. & Lee, L. (2004). A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL '04*, pages 271–278, Morristown, NJ, USA. Association for Computational Linguistics.

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. (2006). Human computing and machine understanding of human behavior: a survey. In *Proc. of ICMI '06*, pages 239–248, New York, USA. ACM Press.

Pascoe, J., Ryan, N., & Morse, D. (2000). Using while moving: HCI issues in fieldwork environments. *ACM Transactions on Computer Human Interaction*, **7**(3), 417–437.

Payne, J. & Bettman, J. (1988). Adaptive strategy selection in decision making. *Learning, Memory, and Cognition*, **14**(3), 534–552.

Pelachaud, C. & Bilvi, M. (2003). Computational model of believable conversational agents. In M.-P. Huget, editor, *Communication in Multiagent Systems*, volume 2650 of *Lecture Notes in Computer Science*, pages 300–317. Springer.

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology*, **54**, 547–577.

Pentland, A. (2000). Perceptual user interfaces: perceptual intelligence. *Communication ACM*, **43**(3), 35–44.

Perlovsky, L. I. (1999). Emotions, learning and control. In *Proc. of International Symposium: Intelligent Control, Intelligent Systems and Semiotics*, pages 131–137, Cambridge, MA.

Pfleger, N., Eger, N. P., Alexandersson, J., & Becker, T. (2002). Scoring functions for overlay and their application in discourse processing. In *Proc. of KONVENS 2002*, pages 139–146, Saarbruecken, Germany.

Picard, R. W. (1997). *Affective Computing*. The MIT Press, Cambridge, Massachusetts.

Pierce, C. S. (1955). Logic as semiotic: The theory of signs. In J. Buchler, editor, *Philosophical Writings of Pierce*, pages 98–119. Dover Publications, New York.

Polanyi, L. & Zaenen, A. (2005). Contextual valence shifters. In J. Shanahan, Y. Qu, and J. Wiebe, editors, *Computing Attitude and Affect in Text*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer.

Potamianos, A., Ammicht, E., & Kuo, H.-K. J. (2000). Dialogue Management in the Bell Labs Communicator System. In *Proc. of ICSLP 2000*, volume 2, pages 603–606, Beijing, China.

Prendinger, H. & Ishizuka, M. (2001). Social role awareness in animated agents. In *Proc. of AGENTS '01*, pages 270–277, New York, USA. ACM Press.

Ramaswamy, S., Rogers, M., Crockett, A. D., Feaker, D., & Carter, M. (2006). WHISPER – service integrated incident management system. *International Journal of Intelligent Control and Systems*, **11**(2), 114–123.

Rector, A. L., Newton, P. D., & Marsden, P. H. (1985). What kind of system does an expert need? In *People and Computers: Designing the interface*, pages 239–247. Cambridge: Cambridge University Press, U.K.

Reeves, B. & Nass, C. (1998). *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press, New York, NY, USA.

Reeves, L. M., Lai, J., Larson, J. A., Oviatt, S., Balaji, T. S., Buisine, S., Collings, P., Cohen, P., Kraal, B., Martin, J.-C., McTear, M., Raman, T., Stanney, K. M., Su, H., & Wang, Q. Y. (2004). Guidelines for multimodal user interface design. *Communication ACM*, **47**(1), 57–59.

Ricoeur, P. (1976). *Interpretation Theory: Discourse and the Surplus of Meaning.* Texas Christian University Press, Fort Worth, TX, USA.

Roberta Catizone, A. S. & Wilks, Y. (2003). Multimodal dialogue management in the COMIC project. In *Proc. of EACL'03 Workshop on Dialogue Systems*, pages 25–34, Morristown, NJ, USA. Association for Computational Linguistics.

Rosenbaum, R., Schumann, H., & Tominski, C. (2004). Presenting large graphical contents on mobile devices - performance issues. In *Proc. of IRMA 2004*, pages 23–26, New Orleans, USA. Press.

Rothkrantz, L. J. M., van Vark, R. J., Peters, A., & Andeweg, N. A. (2000). Dialogue control in the Alparon System. In *Proc. of TSD 2000*, volume 1902 of *Lecture Notes in Computer Science*, pages 333–338, Berlin, Heidelberg. Springer/Verlag.

Rothrock, L., Koubek, R., Fuchs, F., Haas, M., & Salvendyk, G. (2002). Review and reappraisal of adaptive interfaces: towards biologically-inspired paradigms. *Theoretical Issues in Ergonomic Science*, **3**(1), 47–84.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors*, **30**(4), 431–443.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, **39**, 1161–1178.

Ryant, N. & Kipper, K. (2004). Assigning XTAG trees to verbnet. In *Proc. of TAG+7*, pages 194–198, Vancouver, Canada.

Santos, P. J., Baltzer, A. J., Badre, A. N., Henneman, R. L., & Miller, M. S. (1992). On handwriting recognition performance: Some experimental results. In *Proc. of the Human Factors Society 36th Annual Meeting*, pages 283–287, Santa Monica, CA, USA. Human Factors Society.

Schank, R. (1972). Conceptual dependency - a theory of natural language understanding. *Cognitive Psychology*, **3**, 552–631.

Schank, R. (1973). *Computer Models of Thought and Language*. W. H. Freeman, San Francisco.

Schiano, D. J., Ehrlich, S. M., Rahardja, K., & Sheridan, K. (2000). Face to interface: facial affect in (hu)man and machine. In *Proc. of SIG CHI '00*, pages 193–200, New York, USA. ACM Press.

Schmandt, C. (1994). *Voice Communication with Computers*. Van Nostrand Rheinhold, New York, USA.

Schmidt, A., Karlsruhe, U., & Laerhoven, K. V. (2001). How to build smart appliances. *IEEE Personal Communications*, **8**, 66–71.

Schuler, K. K. (2005). *Verbnet: a broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

SFMuseum (1989). The virtual museum of the city of san francisco: San fransisco 9-1-1 dispatch tapes, 17 october, http://www.sfmuseum.org.

Shaikh, M. A. M., Prendinger, H., & Ishizuka, M. (2008). Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Applied Artificial Intelligent*, **22**(6), 558–601.

Sharma, R., Yeasin, M., Krahnstoever, N., Rauschert, I., Cai, G., Brewer, I., MacEachren, A. M., & Sengupta, K. (2003). Speech-gesture driven multimodal interfaces for crisis management. *Proc. of the IEEE*, **91**(9), 1327–1354.

Shieber, S. M. & Baker, E. (2003). Abbreviated text input. In *Proc. of IUI 2003*, pages 293–296, New York, NY, USA. ACM.

Shneiderman, B. (1986). *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Shneiderman, B. (1991). Touch screens now offer compelling uses. *IEEE Software*, **8**(2), 93–94, 107.

Shneiderman, B. (2000). Universal usability. *Communication ACM*, **43**(5), 84–91.

Silverman, L. K. (2002). *Upside-Down Brilliance: The Visual-Spatial Learner*. Deleon Publishing Inc., Denver, USA.

Sinatra, R. (1986). *Visual Literacy Connections to Thinking, Reading and Writing*. Charles C. Thomas, Springfield. IL.

Singhal, A. & Rattine-Flaherty, E. (2006). Pencils and photos as tools of communicative research and praxis. *International Communication Gazette*, **68**(4), 313–330.

Smith, P. A. (1996). Towards a practical measure of hypertext usabilty. *Interacting with Computers*, **8**(4), 365–381.

Smith, S. L. & Mosier, J. N. (1986). Guidelines for designing user interface software. Technical Report ESD-TR-86-278, The MITRE Corporation 1986.

Solso, R. L. (1994). *Cognition and Vision*. Mass, London, England and Cambridge.

Sperber, D. & Wilson, D. (1986). *Relevance: Communication and Cognition*. Harvard University Press, Cambridge, MA, USA.

Steedman, M. & Baldridge, J. (2005). Combinatory categorial grammar. In R. Borsley and K. Borjars, editors, *Non-Transformational Syntax*. Blackwell.

Stock, O. & Zancanaro, M., editors (2005). *Multimodal Intelligent Information Presentation*, volume 27 of *Text, Speech and Language Technology*. Springer, Dordrecht, the Netherlands.

Strapparava, C. & Valitutti, A. (2004). Wordnet-affect: an affective extension of wordnet. In *Proc. of LREC 2004*, pages 1083–1086, Lisbon, Portugal. European Language Resources Association.

Subasic, P. & Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transaction on Fuzzy Systems*, **9**, 483–496.

Svenson, O. & Edland, A. (1987). Change of preferences under time pressure: choices and judgments. *Scandinavian Journal of Psychology*, **28**(4), 322–330.

Tarasewich, P. (2003). Designing mobile commerce applications. *Communication ACM*, **46**(12), 57–60.

Tatomir, B. & Rothkrantz, L. J. M. (2006). Intelligent system for exploring dynamic crisis environments. In B. V. de Walle and M. Turoff, editors, *Proc. of ISCRAM 2006*, pages 288–298, Newark, NJ, USA.

Tennenhouse, D. (2000). Proactive computing. *Communication ACM*, **43**(5), 43–50.

Teo, H.-H., Oh, L.-B., Liu, C., & Wei, K.-K. (2003). An empirical study of the effects of interactivity on web user attitude. *International Journal Human-Computer Study*, **58**(3), 281–305.

Tokuhisa, M., Murakami, J., & Ikehara, S. (2006). Construction and evaluation of text-dialog corpus with emotion tags focusing on facial expression in comics. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4253 of *Lecture Notes in Computer Science*, pages 715–724. Springer, Berlin/Heidelberg, Germany.

Tsutsui, T., Saeyor, S., & Ishizuka, M. (2000). MPML: A Multimodal Presentation Markup Language with character agent control functions. In *Proc. of WebNet 2000*, pages 537–543, San Antonio, USA.

Tufiş, D. & Mason, O. (1998). Tagging Romanian texts: A case study for QTAG, a Language Independent Probabilistic Tagger. In *Proc. of LREC 1998*, pages 589–596, Granada, Spain. European Language Resources Association.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proc. of ACL '02*, pages 417–424, Morristown, NJ, USA. Association for Computational Linguistics.

Turney, P. D. & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transaction on Information System*, **21**(4), 315–346.

USAToday (2001). Wireless tools and long arms of the law (November 30, 2001), http://www.usatoday.com/life/cyber/wireless.

Uther, M. (2002). Mobile internet usability: What can 'mobile learning' learn from the past? In *Proc. of IEEE WMTE '02*, pages 174–176, Washington, DC, USA. IEEE Computer Society.

Valitutti, A., Strapparava, C., & Stock, O. (2004). Developing affective lexical resources. *PsychNology Journal*, **2**(1), 61–83.

van Dam, A. (1997). Post-wimp user interfaces. *Communication ACM*, **40**(2), 63–67.

Vanderdonckt, J., Florins, M., Oger, F., & Macq, B. (2001). Model-based design of mobile user interfaces. In M. Dunlop and S. Brewster, editors, *Proc. of HCI 2001*, pages 135–140, Lille, France.

Vark, R. J. V., de Vreught, J. P. M., & Rothkrantz, L. J. M. (1996). Classification of public transport information dialogues using an information-based coding scheme. In *Proc. of ECAI 1996*, pages 55–69, Budapest, Hungary.

Wahlster, W. (2006). Dialogue systems go multimodal: The smartkom experience. In W. Wahlster, editor, *SmartKom: Foundations of Multimodal Dialogue Systems*, Cognitive Technologies. Springer, Berlin, Heidelberg.

Wallace, R. S. (2003). The elements of aiml style. Technical report, A.L.I.C.E. Artificial Intelligence Foundation, Inc., http://www.alicebot.org.

Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., & Weiser, M. (1995). An overview of the ParcTab ubiquitous computing experiment. *IEEE Personal Communications*, **2**, 28–43.

Ward, D. J., Blackwell, A. F., & MacKay, D. J. C. (2000). Dasher, a data entry interface using continuous gestures and language models. In *Proc. of UIST '00*, pages 129–137, New York, USA. ACM Press.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, **265**(3), 94–104.

Weiss, S. (2002). *Handheld Usability*. John Wiley & Sons, Inc., New York, USA.

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

Whissell, C. M. (1989). The dictionary of affect in language. *Journal of Emotion: Theory, Research and Experience*, **4**, 113–131.

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *Proc. of IAAI 2000*, pages 735–740. AAAI Press / The MIT Press.

Wileman, R. E. (1993). *Visual Communicating*. Educational Technology Publications, Englewood Cliffs, NJ.

Wilson, A. & Oliver, N. (2005). Multimodal sensing for explicit and implicit interaction. In *HCI International 2005*, Mahwah, NJ, USA. Lawrence Erlbaum.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of HLT '05*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

WNBC (2002). Exclusive: 911 tapes tell horror of 9/11 (part 1 and 2): Tapes released for first time, http://www.wnbc.com/news/1315646/detail.html.

Wojdel, A. (2005). *Knowledge Driven Facial Modeling*. Ph.D. thesis, Delft University of Technology, The Netherlands.

Wojdel, J. C. (2003). *Automatic Lipreading in the Dutch Language*. Ph.D. thesis, Delft University of Technology.

Wood, M. E. J. & Lewis, E. (1996). Windmill: The use of a parsing algorithm to produce predictions for disabled persons. In *Proc. of Autumn Conference on Speech and Hearing*, volume 18, pages 315–322.

XTAG-Project (2001). A [lexicalized tree adjoining grammar] for [english], technical report [ircs]-01-03. [ircs]. Technical report, XTAG Research Group, University of Pennsylvania.

Yang, Z. & Rothkrantz, L. J. M. (2007). Dynamic scripting in crisis environments. In N. M. Aykin, editor, *Proc. of HCI International*, volume 4560 of *Lecture Notes in Computer Science*, pages 554–563, Beijing, China. Springer.

Yang, Z., Fitrianie, S., Datcu, D., & Rothkrantz, L. J. M. (2009). An aggression detection system for the train compartment. In A. Solanas and A. Martinez-Balleste, editors, *Advances in Artificial Intelligence for Privacy Protection and Security*, volume 1, pages 249–286. World Scientific Publishing.

ZetaTalk (2003). Chat logs - ZetaTalk Live Dec 2001-May 2003. Retrieved in January 5, 2007, http://www.zetatalk3.com/index/zetalogs.htm.

Zhai, S. & Kristensson, P.-O. (2003). Shorthand writing on stylus keyboard. In *Proc. of SIG CHI '03*, pages 97–104, New York, USA. ACM Press.

Zlango (2009). Pic talk. Technical report, Zlango.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., & Hetherington, L. (2000). JUPITER: A telephone-based conversational interface for weather information. *IEEE Transaction on Speech and Audio Processing*, **8**, 85–96.

Summary

Human Handheld-Device Interaction – An Adaptive User Interface by Siska Fitrianie

The move to smaller, lighter and more powerful (mobile) handheld devices, whether PDAs or smart-phones, looks like a trend that is building up speed. With numerous embedded technologies and wireless connectivity, the drift opens up unlimited opportunities in daily activities that are both more efficient and more exciting. Despite all these advancing possibilities, the shrinking size and the mobile use impose challenges for both technical and usability aspects of the devices and their applications. An adaptive user interface, that is able to autonomously adjust its display and available actions to current goals, contexts and emotions of its user, represents solutions for limited input options, various constraints of the output presentation, and user requirements due to mobility and attention shifting in human handheld-device interaction.

The present work made preliminary steps in proposing a framework for a rapid construction of adaptive user interfaces that are multimodal, context-aware and affective, on handheld devices. The framework consists of predefined modules that are able to work in isolation but can also be connected in an ad hoc way as part of the framework. The modules deal with human handheld-device interaction, the interpretation of the user's actions, knowledge structure and management, the selection of appropriate responses and the presentation of feedback.

Human language and visual perception models have been studied in formulating concepts or ideas as both text and visual language-based messages. An adaptive circular on-screen keyboard and visual language-based interfaces have been proposed as alternative input options for fast interaction. In particular, sentences in the visual language can be constructed using spatial arrangements of visual symbols, such as icons, lines, arrows and ellipses. As icons offer a potential across language barriers, any interaction using the visual language is suitable for language-independent contexts. Personalized predictive and language-based features have also been added to accelerate both input methods.

An ontology has been chosen to represent knowledge of the user, the task and the world. The modeling and structure of the knowledge representation has been designed for sharing common semantics, integrating the communication inter-modules, and fulfilling the context aware requirement. It enables the framework to be developed into a widespread application for different domains. The context awareness is approached by interpreting both verbal and non-verbal aspects of user inputs to update the system's belief about the user, the task and the world. Methods and techniques to fuse multiple input modalities for multiple messages from multiple users into a coherence and context dependent interpretation have been developed. A simple approach to emotion analysis has been proposed to interpret the nonverbal aspect of the inputs. It is based on a keyword spotting approach by categorizing the emotional state into a certain valence orientation with intensity. The approach is suitable for a high uncertainties input recognition.

Template-based interaction management and output generation methods have been developed. The templates have a direct link to concepts in the ontology-based knowledge representation. This approach supports a common semantic with other modules within the framework. It allows the development of a bigger scale system with consistent and easy to verify knowledge repositories.

A multimodal, multi-user, and multi-device communication system in the field of crisis management built based on the framework has been developed as a proof of the proposed concepts. This system consists of comprehensive selected modules for reporting and collaborating observations using handheld devices in mobile ad-hoc network-based communication. It supports communication using the combination of text, visual language and graphics. The system is able to interpret user messages, construct knowledge of the user, the task and the world, and develop a crisis scenario.

User tests were aimed at an assessment of whether or not users are capable of expressing their messages using the provided modalities. The tests also addressed usability issues on interacting with an adaptive user interface on handheld devices. The experimental results indicated that the adaptive user interface is able to support communication between users and between users and their handheld devices. Moreover, an explorative study within this research has also generated knowledge regarding (technical, social and usability aspects of) user requirements in adaptive user interfaces and (generally) human handheld-device interaction. The rationale behind our approaches, designs, empirical evaluations and implications for research on the framework for an adaptive user interface on handheld devices are also described in this thesis.

Samenvatting

Mens-Zakcomputers Interactie – een Adaptieve Gebruikers Interface
door Siska Fitrianie

De beweging naar kleinere, lichtere and krachtigere (mobiele) zakcomputers, PDA's of slimme telefoons, ziet eruit als een trend naar hogere verwerkingssnelheid. Met talloze ingebedde technologieën en draadloze verbindingen, wordt een beweging ingezet naar eindeloze mogelijkheden in dagelijkse activiteiten die zowel efficiënter en opwindender zijn. Ondanks al deze voortschrijdende mogelijkheden, vormen de kleiner wordende afmetingen en het mobiele gebruik, uitdagingen voor technische en gebruikers aspecten van de apparaten en hun toepassingen. Een adaptieve gebruikers interface, die in staat is zelfstandig zijn scherm aan te passen t.b.v. beschikbare acties voor onderhavige doelen, rekening te houden met context en emoties van gebruikers, biedt oplossingen voor beperkte in- en output opties en gebruikers eisen inzake mobiliteit en veranderende aandacht in mens-zakcomputer interactie.

In dit werk worden de eerste stappen gezet naar een raamwerk voor snelle constructie van adaptieve gebruikers interfaces die multimodaal, context gevoelig en effectief zijn in zakcomputers. Het raamwerk bestaat uit voorgedefiniëerde modules die zelfstandig kunnen werken maar ook op een ad-hoc manier gekoppeld kunnen worden. De modules betreffen mens-zakcomputers interactie, de interpretatie van de acties van gebruikers, kennisstructuren en management en de selectie van geschikte antwoorden en presentatie van feedback.

Menselijke taal en visuele perceptiemodellen zijn bestudeerd bij het formuleren van concepten en ideeën t.b.v zowel tekst als visuele op taal gebaseerde. Een adaptief cirkelvormig toetsenbord en visuele op taal gebaseerde interfaces zijn voorgesteld als alternatieve input opties voor snelle interacties. In het bijzonder kunnen zinnen in de visuele taal geconstrueerd worden door gebruik te maken van ruimtelijk geordende visuele symbolen zoals iconen, lijnen, pijlen en ellipsen. Omdat iconen de mogelijkheid bieden taalbarriéres te overschrijden, is iedere interactie met behulp van visuele taal geschikt voor gebruik in taalonafhankelijke contexten. Persoonafhankelijke voorspelling en op taal gebaseerde kenmerken zijn ook toegevoegd om beide input methodes te versnellen.

Een ontologie is gekozen om de kennis van de gebruiker, de taak en de wereld te representeren. De kennis representatie is ontworpen om gemeenschappelijke semantiek, de communicatie tussen modules te integreren en om de context bewustwordingseis te vervullen. Het geeft mogelijkheden om het raamwerk te ontwikkelen voor een veelvoud van applicaties in verschillende domeinen. De context gevoeligheid is benaderd door de interpretatie van de gebruiks-taak en -omgeving te actualiseren. Er zijn methoden en technieken ontwikkeld om multiple input modaliteiten te fuseren van meervoudige boodschappen en van meerdere gebruikers, tot een coherente en context afhankelijke interpretatie. Een simpele manier voor analyse van emoties is voorgesteld om de non-verbale aspecten van de input te kunnen interpreteren. Het is gebaseerd op het vinden van sleutelwoorden om de emotionele toestanden positief-negatief te kunnen categorieën. Deze aanpak herkent zelfs zeer onwaarschijnlijke input.

Op interactiestandaards zijn management en output genererende methoden ontwikkeld die directe gekoppeld zijn aan de ontologie van de kennisrepresentatie. Deze benadering ondersteunt een gemeenschappelijke semantiek met andere modules binnen het raamwerk. Dit maakt de ontwikkeling van een grooter systeem met consistente en gemakkelijk te verifiëren kennis bestanden.

Een multimodaal communicatie systeem met meerdere gebruikers en meerdere apparaten voor het crisisdomein is ontwikkeld als een proof of concept. Het systeem bestaat uit een omvangrijke, selectie van modules om samenhangende observaties te rapporteren door gebruik te maken van zakcomputers in mobiele ad-hoc netwerken gebaseerde communicatie. Het ondersteunt de communicatie door gebruik te maken van een combinatie van tekst, visuele en grafische taal. Het systeem is in staat om boodschappen van gebruikers te interpreteren, kennis van de gebruiker 'construeren en zowel taken als een geheel crisisscenario te genereren.

Gebruikstesten waren gericht op het bepalen of de gebruikers in staat waren hun boodschap uit te drukken met behulp van de aangeboden modaliteiten. De testen behandelen ook bruikbaarheid inzake interactie met adaptieve gebruikers interfaces op zakcomputers. De experimentele resultaten laten zien dat de adaptieve gebruikers interface in staat is om communicatie tussen gebruikers and tussen gebruikers en hun zakcomputers te ondersteunen. Bovendien heeft een exploratieve studie binnen dit onderzoek ook kennis gegenereerd met betrekking tot (technische, sociale en bruikbaarheidsaspecten) van gebruikers eisen in adaptieve gebruikers interface en (algemene) mens-zakcomputer interactie. De rationale achter al deze benaderingen, ontwerpen, empirische evaluaties en implicaties voor onderzoek naar een raamwerk voor een adaptieve gebruikers interface voor zakcomputers zijn ook beschreven in deze proefschrift.

Curriculum Vitae

Siska Fitrianie was born in Bandung, Indonesia, on September 22nd, 1978. After graduating from SMAN 3 Bandung in 1995, she pursued Informatics Engineering at Bandung Institute of Technology. Her bachelor thesis project was in the field of Inductive Logical Programming (Machine Learning). In fall 2000, Siska was awarded the STUNED scholarship (from the Netherlands Education Center) to Delft University of Technology, the Netherlands, where she studied Technical Informatics with specialization Artificial Intelligent. With her thesis work "My Eliza, a Multimodal Communication System", she obtained her master degree with honor in 2002. For the next two years, Siska completed a post-graduate programme at User System Interaction, Eindhoven University of Technology and received her Professional Doctoral degree in Engineering with her thesis work "An Icon-based Communication Tool on a PDA" in 2004.

From September 2004, Siska was involved in the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, as her PhD project at the Man-Machine-Interaction group, Delft University of Technology. In the fourth year of the PhD project, she received the 2008 Google Anita Borg Memorial Scholarship. Her research was related with modeling processing and interaction solutions for managing the interaction between humans and the interfaces of computer systems. This thesis reflects her research work carried out during these years.

Publications

Fitrianie, S., Yang, Z., Datcu, D., Chiţu, A. G., & Rothkrantz, L. J. M. (2010). Context-aware Multimodal Human-Computer Interaction. In R. Babuska and F. C. A. Groen, editors, *Interactive Collaborative Information Systems*, volume 281 of *Studies in Computational Intelligence*, pages 237–272. Berlin, Heidelberg, Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2009). Computed Ontology-based Situation Awareness of Multi-User Observations. In J. Landgren and S. Jul, editors, *Proc. of ISCRAM 2009*, Gothenburg, Sweden.

Fitrianie, S. & Rothkrantz, L. J. M. (2009). Database for a Visual Language-based Application. In *Proc. of Euromedia 2009*, pages 10–17, Bruges, Belgium. Eurosis.

Yang, Z., Fitrianie, S., Datcu, D., & Rothkrantz L. J. M. (2009). An Aggression Detection System for the Train Compartment. In A. Solanas and A. Martinez-Balleste, editors, *Advances in Artificial Intelligence for Privacy Protection and Security*, no. 1, pages 249–286. World Scientific Publishing Co.

Fitrianie, S. & Rothkrantz L. J. M. (2008). An Automated Online Crisis Dispatcher. *International Journal of Emergency Management*, 5(1–2), 123–144.

Fitrianie, S. & Rothkrantz L. J. M. (2008). The Generation of Emotional Expressions for a Text-Based Dialogue Agent. In *Proc. of TSD 2008*, volume 5246 of *Lecture Notes in Computer Science*, pages 569–576, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2008). A Language-Independent Application using Icon Language. In *Proc. of ICTS 2008*, pages 627-633, Surabaya, Indonesia.

Fitrianie, S., Tatomir, I., & Rothkrantz L. J. M. (2008). A Context Aware and User Tailored Multimodal Information Generation in a Multimodal HCI Framework. In *Proc. of Euromedia 2008*, pages 95–103, Proto, Portugal. Eurosis.

Fitrianie, S., Yang, Z., & Rothkrantz, L. J. M. (2008). Developing Concept-Based User Interface using Icons for Reporting Observations. In F. Friedrich and B. van de Walle, editors, *Proc. of ISCRAM 2008*, pages 394–403, Washington, DC, USA.

Fitrianie, S., Datcu., D., & Rothkrantz, L. J. M. (2007). Human Communication Based on Icons in Crisis Environments. In N. M. Aykin, editor, *Proc. of HCI International 2007*, volume 4560 of *Lecture Notes in Computer Science*, pages 57-66, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2007). An Adaptive Keyboard with Personalized Language-based Features. In *Proc. of TSD 2007*, volume 4629 of *Lecture Notes in Computer Science*, pages 131–138, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2007). An Automated Crisis Online Dispatcher. In B. v. d. Walle, P. Burghardt, and K. Nieuwenhuis, editors, *Proc. of ISCRAM 2007*, pages 525–536, Delft, the Netherlands.

Fitrianie, S. & Rothkrantz, L. J. M. (2007). An Automated Text-Based Synthetic Face with Emotions for Web Lectures. *Communication and Cognition - Artificial Intelligence*, 23(1–4), 89–98.

Fitrianie, S. & Rothkrantz, L. J. M. (2007). A Visual Communication Language for Crisis Management. *International Journal of Intelligent Control and Systems, Distributed Intelligent Systems*, 12(2), 208–216.

Fitrianie, S. & Rothkrantz, L. J. M. (2007). Personalized Adaptive PDA Interface. In *Proc. of Euromedia 2007*, pages 98–105, Delft, the Netherlands. Eurosis.

Fitrianie, S., Poppe, R., Bui, T. H., Chiţu, A. G., Datcu, D., Dor, R., Hofs, D. H. W., Wiggers, P., Willems, D. J. M., Poel, M., Rothkrantz, L. J. M., Vuurpijl, L. G., & Zwiers, J. (2007). A Multimodal Human-Computer Interaction Framework for Research into Crisis Management. In v. d. B. Walle, P. Burghardt, and K. Nieuwenhuis, editors, *Proc. of ISCRAM 2007*, pages 149–158, Delft, the Netherlands.

Fitrianie, S., Datcu, D., & Rothkrantz, L. J. M. (2006). Constructing Knowledge of the World in Crisis Situations using Visual Language. In *Proc. of IEEE SMC 2006*, pages 121–126. IEEE Computer Society. (The best paper award)

Fitrianie, S. & Rothkrantz, L. J. M. (2006). Constructing Knowledge for Automated Text-Based Emotion Expressions. In *Proc. of CompSysTech 2006*, pages (V.6) 1–6, Veliko Tarnovo, Bulgaria. (The best paper award)

Fitrianie, S. & Rothkrantz, L. J. M. (2006). A Text-Based Synthesis Face with Emotions. In *Proc. of Euromedia 2006*, pages 28–32, Athens, Greece. Eurosis.

Fitrianie, S. & Rothkrantz, L. J. M. (2006). Two-Dimensional Visual Language Grammar. In *Proc. of TSD 2006*, volume 4188 of *Lecture Notes in Computer Science*, pages 573–580, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. & Rothkrantz, L. J. M. (2005). Communication in Crisis Situations using Icon language. In *Proc. of IEEE ICME 2005*, pages 1370–1373, Amsterdam, the Netherlands. IEEE Computer Society.

Fitrianie, S. & Rothkrantz, L. J. M. (2005). An Icon-Based Communication Tool on a PDA. In *Proc. of Euromedia 2005*, pages 83–90, Tolouse, France. Eurosis. (The best paper award)

Fitrianie, S. & Rothkrantz, L. J. M. (2005). Language-Independent Communication using Icons on a PDA. In *Proc. of TSD 2005*, volume 3658 of *Lecture Notes in Computer Science*, pages 404–411, Berlin, Heidelberg. Springer-Verlag.

Rothkrantz, L. J. M., Datcu., D., Fitrianie, S., & Tatomir, B. (2005). Personal Mobile Support for Crisis Management using Ad-Hoc Networks. *The 11th International Conference on Human-Computer Interaction*. Lawrence Erlbaum Associates Inc.

Fitrianie, S. (2004). *An Icon-based Communication Interface on a PDA*. Professional Doctoral in Engineering's thesis, Post-Graduate Programme of User System Interaction, Eindhoven University of Technology.

Weevers, I., Sluis, R. J. W., van Schijndel, C. H. G. J., Fitrianie, S., Kolos-Mazuryk, L., & Martens, J. -B. O. S. (2004). Read-It: A Multi-modal Tangible Interface for Children Who Learn to Read. In *Proc. of ICEC 2004*, volume 3166 of *Lecture Notes in Computer Science*, pages 226–234, Berlin, Heidelberg. Springer-Verlag.

Sluis, R. J. W., Weevers, I., van Schijndel, C. H. G. J., Kolos-Mazuryk, L., Fitrianie, S., & Martens J. -B. O. S. (2004). Read-It: Five-to-Seven-Year-Old Children Learn to Read in a Tabletop Environment. In *Proc. of IDC 2004*, pages 73–80, Maryland, USA. ACM Press.

Fitrianie, S. & Rothkrantz, L. J. M. (2003). My_Eliza, A Multimodal Communication System. In *Proc. of Euromedia 2003*, pages 14–22, Plymouth, UK. Eurosis. (The best paper award)

Fitrianie, S., Wiggers, P., & Rothkrantz, L. J. M. (2003). A Multi-modal Eliza using Natural Language Processing and Emotion Recognition. In *Proc. of TSD 2005*, volume 2807 of *Lecture Notes in Computer Science*, pages 394–399, Berlin, Heidelberg. Springer-Verlag.

Fitrianie, S. (2002). *My_Eliza, A Multimodal Communication System*. Master's thesis. Delft University of Technology.

Fitrianie, S. (2000). *A Workbench of Inductive Logic Programming Algorithms*. Bachelor's thesis. Bandung Institute of Technology.