MUSCLE

Network of Excellence

Multimedia Understanding through Semantics, Computation and Learning

Project no. FP6-507752

Deliverable D.10.1

State-of-the-Art Report on

Human Computer Interfaces for Multimedia Retrieval

Due date of deliverable: 01.09.2004
Actual submission date: 01.09.2004

Start date of Project: 1 March 2004 Duration: 48 Months

Name of responsible editor(s):

- Manolis Perakakis, Michail Toutoudakis, Alexandros Potamianos (TUC);
- Santtu Toivonen, Sanni Siltanen, Seppo Valli, Juha Leppanen (VTT);
- Fred Stentiford, Ingemar Cox (UCL);
- George Papandreou, Nassos Katsamanis, Petros Maragos (NTUA)

Revision: 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including Commission Services)	
RE	Restricted to a group specified by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

Keyword List:

Network of Excellence FP6-5077-52
Multimedia **U**nderstanding through **S**emantics, **C**omputation and **Le**arning

WP10 : A State of the Art Report on Human Computer Interfaces for Multimedia Retrieval

Edited by :
Manolis Perakakis, Michail Toutoudakis and Alexandros Potamianos(TUC)
Santtu Toivonen, Sanni Siltanen, Seppo Valli, Juha Leppanen (VTT)
Fred Stentiford, Ingemar Cox (UCL)
George Papandreou, Nassos Katsamanis and Petros Maragos (NTUA)

September 2004

Contents

List of Figures

List of Tables

1 Introduction

In this review paper we are investigating the state of the art in the technologies involved in the field of *Human Computer Interfaces for Multimedia Retrieval.* In the first section, we present a state-of-the-art review of HCI. First we define the area of *Human Computer Interaction* HCI and give a brief historical overview of the most important breakthroughs seen as usability paradigms of the domain. Then we analyze the components of HCI, namely the user, the system and the interaction. In the analysis of the system we introduce the main input/output components and modalities of the system. Then we introduce the main styles of interaction that take place between the computer and the end user. At the end of this chapter we present research issues and challenges of HCI technology.

In Chapter 3, we review *Graphical User Interfaces* (GUIs); GUIs dominate today's user interfaces. We begin by giving the definition of the GUI area of research. We then present the main GUI design principles and guidelines. Afterwords we present the X-Window system case study in order to show the layered architecture of graphical user interfaces. Then a historical review on the various tools used to create GUIs is presented; a distinction between successful and unsuccessful tools is included. Finally in this chapter, we include the information exploration and visualization subsection. In the information visualization section, we point out the importance of this task and introduce the *data type by taxonomy* schema for information visualization.

In Chapter 4, we investigate the speech and natural language interfaces based on the technologies of Automatic Speech Recognition, Text-to-Speech Synthesis, Natural Language Understanding and Natural Language Generation. The technology is introduced and its benefits are outlined; a historical reviews of speech applications is also included. Then we analyze the effects of speech technology on system application design. The state of the art of the main components of spoken language interfaces, namely speech recognition (ASR), speech synthesis (TTS), semantics and spoken language understanding is also presented.

In Chapter 5, we review efforts in defining design principles and creating tools for building multimodal dialog systems with emphasis on the speech modality. General design principles for architecting and building such systems are reviewed and challenges are outlined. The focus is on system architecture, application and speech interface design, data collection and evaluation tools. We conclude that modularity, flexibility, customizability, domain-independence and automatic dialog generation are some important features of successful dialog systems and design tools.

Next, we turn our attention to the very active field of multimodal systems, which uses a multiple of modalities (speech, gesture, etc) to augment user's interaction with the system. First we provide motivation for the use of multimodal interfaces and briefly review some examples of multimodal applications. Then we examine topics such as multimodal interaction (mainly fusion and integration techniques) and dialog management handling strategies. In the dialog management section, we focus on dialog manager initiative strategies namely system initiative, user initiative and mixed initiative dialog strategies. Then a brief presentation of the dialog management models used by various systems to control the dialog is presented, namely finite states machine and frame-based strategies. Finally we examine issues regarding to system architecture and conclude with various efforts for standardizing multimodal interaction specially on the web.

Special issues like Eye Tracking and Audio-Visual Speech Recognition are discussed in the following chapters. We begin with an introduction to the technology and the methods that have been developed over the years. Then a review of the human gaze behavior is presented analyzing various experiments that have been conducted to explore human gaze behavior for different purposes. A state of the art review of Eye Tracking Interfaces is included. Finally we outline the main shortcomings of the eye tracking technology. Audio-Visual Speech Recognition is discussed in the next chapter. Multimodal feature extraction and multimodal feature fusion are two main areas investigated in this review.

Adaptive user interfaces are discussed in Chapter 9. We start by motivating the work and giving the definition of adaptive user interfaces. Then we discuss the nature of adaptive interfaces by giving a reference to *rapid and online* learning. Next we define the main categories of adaptive user interfaces namely informative, generative and conversational systems based mainly on the of communication between the system and the end user. Next, we provide a section with the main benefits and limitations of such systems followed by the main approaches for designing adaptive user interfaces. We conclude this chapter with an outline of the most important objective measures in the evaluation of various types of adaptive user interfaces.

In the final chapter of this state-of-the-art review, mobile interfaces are presented. Mobile content should be

easy to create and access with mobile devices which have limited capabilities for input, output, storage, memory, processing power, etc. Novel methods for overcoming these difficulties are presented. Alternative interfaces are presented, as well as semantics and meta-data for efficient storage and retrieval of multimedia content. Some comparisons between capabilities of handheld devices and those of "traditional" PCs are also considered.

This paper is by no means an exhaustive review of the area of human-computer interfaces. This state-of-the art review mainly concentrates on areas and cutting-edge technologies that are most interesting and relevant for the MUSCLE NoE project, and specifically by WP10 participants, i.e., interfaces for multimedia information retrieval.

2 Human Computer Interaction

Human-computer interaction is the study, planning and design of how people and computers work together so that a person's needs are satisfied in the most effective way. We study HCI to determine how we can make computer technology more usable by people, which requires an understanding of at least four things: the computer technology, the people who interact with it, what is meant by *more usable* and also the understanding the work people are trying to accomplish by using computer technology.

HCI is multi-disciplinary subject, since a designer of an interactive system would have expertise in range of topics: *psychology and cognitive science* to give knowledge of user's perceptual, cognitive and problem solving skills, *sociology* to help the user understand the wider context of interaction, *ergonomics* for the user's physical capabilities, *graphic design* to produce effective interface presentation, *computer science and engineering* to be able to build necessary technology, etc. The question of whether HCI is a science or a craft discipline is an interesting one. Does it involve artistic skill and fortuitous insight or methodological science? Drawing an analogy with architecture, beautiful and novel interfaces are artistically pleasing and capable of fulfilling the tasks required. However, interface efficiency and usability are equally important.

This chapter, is an introduction to HCI. First, we present a brief history of HCI and then analyze its three components, namely the user, the system and the interaction and we conclude with some interesting research issues and challenges. We address special issues in the following chapters.

2.1 Usability paradigms in HCI history

Before starting developing some of the elements of theory of HCI, we give a brief historical overview, with some of the more important breakthroughs seen as usability paradigms.

One of the first advances in computer interaction is the transition from the early batch systems to *multi-user time-sharing systems*. By the 1960s, it was becoming apparent that the explosion of growth in computer power would be wasted if there was not an equivalent explosion of ideas about how to channel that power. Previously users(programmers) were restricted to batch sessions, in which complete jobs were submitted on punched cards to operators who would run them on the computer. Rather than rely on a model of interaction as a pre-planned activity, truly interactive exchange between programmer and computer was now possible. Computers became dedicated partners with each individual user and allowed the human to become a more reactive and spontaneous collaborator.

By mid-1950s researchers experimented with the possibility of presenting and manipulating information from a computer in the form of images on a video display unit (VDU). It was 1962 when Ivan Sutherland a graduate student of MIT, astonished the computer science community with his *Sketchpad* program ([303], [304]) that the capabilities of visual images was realized. Sketchpad was not just a tool for creating visual displays, it was a kind of simulation language and a model for totally new ways of using computers by changing something on the display screen. Sketchpad demonstrated that computers could be used to extend user's ability to abstract away from some levels of detail by visualizing and manipulating different representations of the same information. It also showed how important the contribution of one creative mind could be to the entire history of computing.

Douglas Engelbart's ambition at Berkeley was to use computers for teaching. The dream of naive users learning from computers sound at least ambitious that times. Many of his ideas became reality during the development of the NLS/Augment system at *Standford Research Institute* (SRI) during 1960s, with the invention of word-processing and the mouse [105]. But building complex interactive systems required the use of appropriate tools. Software programs servicing that role became a reality, giving the idea of *software toolkits*. In 1970s Seymour Papert at MIT, wanted to develop a language that was easy for children to use thus inventing *LOGO* language, demonstrating that no matter how powerful a system may be, it will always be more powerful the easier it is to use.

Alan Kay, influenced by Engelbart and Papert realized that the power of a system such as NLS was only going to be successful, if it was accessible by novice users as was LOGO. His view of the future of computing was embodied in small, powerful machines which were dedicated to single users, that is *personal workstations*. Kay and a team from Xerox Palo Alto Research Center (PARC) worked on incorporating a powerful and simple visually based programming environment, *Smalltalk*, for the personal computer that was just becoming feasible. Kay's vision extended to handheld personal computing with *Dynabook*, a portable interactive personal computer,

as accessible as a book (what is now known as a laptop/tablet pc).

The Xerox Alto and Star (1981) were the first personal workstations having significant local processing power and memory, networking capabilities, a high resolution bit-mapped display, a keyboard and a mouse. The user interface incorporated windows, menus, scrollbars, mouse control and selection mechanisms (*WIMP* interface - windows, icons, menus and pointers) and views of abstract structures all presented in a consistent manner. These systems introduced several innovative concepts found in todays personal computers: the *desktop metaphor*, *direct manipulation* and WYSIWYG -*what you see is what you get*-in which a user sees and manipulates in the screen a representation of a document that looks identical to the eventual printed page.

Vannevar Bush in 1945, then the highest ranking scientific administrator in US, realized that it was becoming extremely difficult to keep in touch with the growing body of the scientific knowledge in the literature. To this end, he described in his article *As we may think* [67] an innovative and futuristic information storage and retrieval apparatus -*the memex*- which was constructed with technology wholly existing in 1945 and aimed at increasing the human capacity to store and retrieve connected pieces of knowledge my mimicking the ability to create random associative links. Stored information in memex would resemble a vast interconnected mesh of data, similarly to how many perceive information is stored in the human brain. Later, Ted Nelson's quest to produce *Xanadu*, a revolutionary worldwide publishing and retrieval system based on the idea of interconnected non-linear text(*hypertext*) and other media forms, finally took place with the creation of the World Wide Web (WWW) by Tim Berners-Lee in 1990s.

2.2 Foundations of HCI: the user, the system and the interaction

In the following three sections we analyze the three components of HCI namely the user, the system and the interaction between them.

2.3 The user

The study of human in the context of HCI draws mainly from *cognitive psychology*. In order to design something, we need to understand the capabilities and limitations of humans: how they perceive the world around them, how they store and process information and solve problems. The *Human Model Processor* (described in [73]) is a simplified view of the human processing involved in interacting with computer systems. The model comprises three subsystems, namely: the perceptual system handling sensory stimulus from the outside world, the motor system which controls actions and the cognitive system which provides the necessary processing to connect the two.

Retaining the analogy of the user as an information processing system but making the analogy closer to a user of a conventional computer system, the model components would include the input-output, memory and processing systems. Study of input-output channels (vision, hearing, touch and movement), human memory (sensory, short-term and working or long-term memory) and processing capabilities (reasoning, problem solving and skill acquisition) should all be considered when designing computer systems with usability in mind. Many studies in literature analyze each subsytem in detail but as such an analysis wouldn't fit here, the interested user should check [95].

2.4 The system

The remarkable progress of the latest years in computer processors speed, memory and storage capabilities is matched by improvement in many input and output devices. We mainly focus on most common Input/Output devices and modalities in this paragraph.

2.4.1 Keyboard

The primary mode for textual data entry is still the *keyboard*. By the 1870's Christopher Latham Sholes invented a good mechanical design and a clever placement of the letters by putting frequently used letter pairs far apart, thereby increasing finger travel distances. This layout's success led to widespread standardization (QWERTY layout). Various other attempts were made for alternative layouts such as the *Dvorak* layout developed in 1920

or the *ABCDE style* which had all the 26 letters in alphabetical order. However the widespread use of the *QWERTY keyboard* standardized it.

2.4.2 Pointing Devices

In graphical user interfaces, the user points at and selects items. This results faster performance, fewer errors, easier learning and higher user satisfaction. According to ([68, 72]) the diversity of tasks and variety of devices plus strategies for using them, create a rich design space. Pointing devices are applicable in six types of interaction [111]: *select, position, orient, path, quantify and text.*

Pointing devices are grouped in two categories according to what they offer: a) direct control on the screen surface such as *light pen, touch screen and stylus*, b) indirect control away from the screen surface such as *mouse, track ball, joystick, graphics tablet and touch pad.*

Direct control pointing devices

The *light pen* was an early device that allowed users to perform all six tasks (see [111]) but also had some disadvantages: user's hand obscured part of the screen; users had to remove their hand from the keyboard and to pick up the pen. The *stylus* is attractive to designers because it permits high precision with good control to limit inadvertent selection especially in small screens such as the PDAs. Alternatives to keyboard entry are gaining popularity with *touch screens* or *stylus* entry on virtual keyboards, but these mechanisms have data-entry rates of only 20 to 30 words per minute according to[285].

Indirect Control Pointing Devices

Indirect control pointing devices eliminate the hand fatigue and screen obscuring problems but must overcome the problem of indirection. As with *light pen* the off-keyboard hand position and pick up problems still remain. Also, indirect control devices require more cognitive processing and hand-eye coordination to bring the on screen cursor to the desired position. The *mouse* is appealing because the hand rests in a comfortable position, buttons on the mouse are easily pressed, even long motions can be rapid and positioning can be precise. It was developed at SRI in 1965 as part of the NLS project to be a cheap replacement of *light-pens* which had been used at least since 1954 [120]. The *trackball* has sometimes been described as an upside-down mouse. It has been the preferred device in the high-stress world of air-traffic control and in some video games. The *joysticks* long history began in aircraft-control devices. Joysticks are appealing for tracking purposes. *Graphics tablet* is a touch sensitive surface separate from the screen, usually laid flat on the table or in the user's lap. A *touchpad* built in near the keyboard offers the convenience and precision of a touchscreen while keeping the user's hand off the display surface.

2.4.3 Output Devices

The primary source of feedback from the computer to the user is the visual display unit (VDU) ([70, 106, 31]). Display technologies include: raster-scan cathode-ray tube (CRT), liquid-crystal displays (LCD), plasma panels and Light-emitting diodes(LEDs). Through the visual display unit, feedback can be represented to the user in the form of text, graphics and video.

2.5 The interaction

Interaction can be seen as a dialogue of the user and the computer. The choice of the interaction style can have a profound effect on the nature of this dialogue. We briefly review most of them in the first part of this section and then we address the topic of interaction models.

2.5.1 Styles of interaction

The *command line interface* was the first interactive dialogue style to be commonly used and despite of the availability of menu driven interfaces, it is still widely used. Command line interfaces are powerful in that they

offer direct access to system functionality and can be combined to apply a number of tools to the same data. They are also flexible, since most commands have a large number of options and parameters which vary its behavior and can be applied at many objects at once. However this power and flexibility brings with it difficulty in use and learning, so they are better for expert users.

In *menu-driven interface*, the set of options available to the user is displayed in the screen and selected using the mouse or keyboard. Since the options are visible, they are less demanding on the user relying on recognition rather than recall. However menu options should be meaningful and logically grouped to aid recognition. Often menus are hierarchically ordered and the option required may not be available in top layer of the hierarchy. Grouping and naming of menu options then provides the cues for the user or find the required option.

Perhaps the most attractive means of communicating with computers is by *natural language*. Natural language understanding, both of speech and written input is the subject of much interest and research. Because language is ambiguous at a number of levels it seems unlikely that a general language interface will be available for some time, if at all. However, systems can be built to understand restricted subsets of a language. For a known and restricted domain, the system can be provided with sufficient information to disambiguate terms. However the user must learn which phrases the computer understands. But even if general language interface ever become available, it is questionable how useful such an interface would be, since language is by nature vague and imprecise -this gives its flexibility and allows creativity in expression- while computers require precise instructions!

Question and answer dialogue is a simple mechanism for providing input to an application in a specific domain. The user is asked a series of questions and so is led through the interaction step by step. These interfaces are easy to learn and use but are limited in functionality and power. As such they are appropriate for restricted domains and novice users. *Query languages* on the other hand are used to construct queries to retrieve information from databases. They use natural language style phrases but in fact require special syntax as well as knowledge of the database structure, so it is useful mainly for experienced users.

The design of a *visual representation* of the world of action can greatly simplify user's tasks because direct manipulation of familiar objects is possible. Examples of such systems include the *popular desktop metaphor*, CAD tools and video games. By pointing at visual representations of objects and actions, users can carry out tasks rapidly and can observe results immediately. *Direct manipulation* is appealing to novices and easy to remember.

2.5.2 Interaction models : GOMS and OAIM

In the early seventies Card, Moran and Nevell (1983) [73] from Xerox PARC Laboratories developed a series of models and condensed them to the Goals, Operators, Methods and Selection (GOMS) rules model for analyzing routine human computer interactions. The model explicitly integrated many components of skilled performance to produce predictions about real tasks. The model was both a great advance to prior human factors modeling and on cognitive psychology and set a standard for scientific and theoretical accuracy and innovation.

As GUIs have replaced command languages, intricate syntax has given way to relatively simple direct manipulations applied to visual representation of objects and actions. By now objects and actions have become dominant features. The underlying theory of design is called *object action interface* model (OAI). The emphasis now is on the visual display of user task objects and actions. Object action design starts with understanding the task, which includes the real-world objects with which users work to accomplish their intentions and the actions they apply to these objects. Once these objects and actions are defined the designer can create the metaphoric representation of the interface objects and actions. OAI model is an explanatory model that focuses on task objects and actions and on interface objects and actions. It also reflects the high level of design with which most designers deal when they use the widgets in user interface-building tools. OAI model is in harmony with the software engineer trends towards object-oriented design and programming methods that have become popular the past decade.

2.5.3 Novel devices and modalities

Quests and needs for new ways to interact with computers has led to the creation of various novel devices and modalities. The future of computing is likely to include novel modalities such as *gestures, speech, haptics*, and

innovative devices, sensors, opening the door to new applications ([291, 124, 110, 72, 152]).

Speech is considered the most natural form of communication and although there are several limitations when used in interaction with computers much progress in both system architectures and applications have been noticed. Results of empirical studies show that animated characters may have a strong motivational impact, since they are considered as being more lively and engaging for many users [191, 315]. *Glove mounted* devices [295] and *graspable user interfaces* seem ripe for exploration ([118, 117]). Pointing devices with *haptic feedback*, *eye-tracking* and *gaze-detection* [153] are also a field of scientific interest.

With the increase of use of portable devices such as PDAs, tablet computers and mobile phones the problem of *text entry* arisen. Currently, there are many types of text entry methods [197] for mobile devices such as *reduced physical keyboards*, *handwriting recognition* ([205, 49, 149]), *word level* [198] and *virtual keyboards* ([336, 335, 337])

2.6 Research issues and challenges

With the appearance of a vast array of computational devices such as phones, embedded systems, PDAs, laptops desktops, wall size displays and various other devices with different sizes, computational power and input/output capabilities the idea of ubiquitous computing [326] is becoming a reality. An important issue is how the interaction techniques should change to take these varying input and output hardware into account (e.g., using a stylus instead of a mouse). The system might choose the appropriate interaction techniques taking into account input and output capabilities of the devices and user preferences. So nowadays many researchers are focusing on such fields as *context aware interfaces, recognition based interfaces, intelligent and adaptive interfaces*, and *multimodal-perceptual interfaces.*

Most tasks users perform are not isolated from the interrelated conditions but occur in certain context. Context can be considered as knowledge of the environment, location, situation, user, or current task. Context awareness can be exploited in selecting application or information, adjusting communication and adapting user interface according to current context. One field where context aware interfaces show a lot of promise is in portable mobile devices such as PDAs and mobile phones through the use of sensors (e.g., location awareness).

Multimodal (or perceptual) interfaces are interfaces where the communication between the system and the user is made thought various modalities such as speech, gesture recognition, eye gaze, haptics, GUI, etc. These systems should have the ability to fuse the information through the various modalities, to decide in each point of the interaction which one is the best modality to communicate with the end user (*adaptive interfaces*) taking into account various features at each time, and to be able to disambiguate input from a modality with the use of an other one (*intelligent interfaces*). Multimodal interfaces differ from today's interfaces in that input is uncertain due to recognizer errors. Therefore interfaces must contain facilities to allow the user to monitor and correct the interpretation, handle errors and interruption effectively and dynamically adapt to the current context and situation.

3 Graphical User Interfaces

In this chapter, we provide a review on Graphical User Interfaces(GUIs). We first define the GUI term, give motivations for GUI usage and present some design principles/guidelines a graphical user interface should fulfill. We present a brief review of the architecture and main components of GUI systems and a review on the development tools (successful and unsuccessful according to [222]) used so far and conclude with a discussion of information exploration and visualization techniques.

3.1 Definition

According to the Computing Dictionary *GUI is the use of pictures rather than just words to represent the input and output of a program.* A program with a GUI runs under some windowing system (e.g. The X Window System, MacOS, Microsoft Windows). The program displays certain icons, buttons, dialogue boxes, etc. in its windows on the screen and the user controls it mainly by moving a pointer on the screen (typically controlled by a mouse) and selecting certain objects by pressing buttons on the mouse while the pointer is pointing at them. This contrasts with a command line interface (CLI) where communication is exchanged by strings of text.

3.2 Motivation

Graphical user interfaces are the applications window to the world. They often define the success of applications in the real world. Software products ranging from simple web-sites to complex molecular visualization tools, all need to provide usable, intuitive user interfaces. Recent technologies like speech and gesture recognition are still not mature and usable enough. Even when they do, they will complement GUIs rather than replace them because of the high bandwidth GUIs provide. Information can be better organized and presented to the end user using GUIs.

3.3 Design Principles/Guidelines

In the design of HCI it is important that the system should achieve the following goals:

- *Proper functionality*: For this purpose task analysis is central; systems with inadequate functionality frustrate the user and are often rejected or underutilized. If the functionality is inadequate it does not matter how well the interface is designed.
- *Reliability, Availability and Data integrity*: A vital step is ensuring system reliability. Commands must function as specified, displayed data must reflect the database contents and updates must be applied correctly.
- *Standardization, integration, consistency, portability*: Standardization reflects to common user interface features across multiple applications. Integration across application packages and software tools which was one of the key design principles of UNIX. Consistency primarily refers to common action sequences, terms, units, layouts, color, typography and so on within an application program. Consistence is a strong determinant of the success of a system. Portability refers to the potential to convert data and to share user interfaces across multiple software and hardware environments.

Multiple design alternatives must be evaluated for specific user communities and for specific benchmark tasks which is the basis of human-factors goals. The five measurable human factors are central to evaluation:

- *Time to learn*: How long it will take typical members of the target community to learn how to use the command relevant to a set of task.
- *Speed of performance*: How long it takes to carry out the benchmark tasks
- *Retention over time*: How well do users maintain their knowledge after some specific time
- *Rate of errors by user*: How many and what kind of errors do people make.

- *Subjective satisfaction*: How much users like or not the various aspects of the system.

The overall goal is to design usable systems that support specified users in their specified context to reach their particular goals effective, efficient and with satisfaction (see ISO 9241-11). Furthermore the dialog has to be designed in the sense that it must be:

- suitable for the task
- self-descriptive
- controllable
- conform with use expectations
- error tolerant
- suitable for individualization
- suitable for learning

In the design of HCI it is important that "presentation of visual information enables the user to perform perceptual task (e.g. searching for information on the screen) effectively, efficiently and with satisfaction" (ISO 9241-12). Following requirements for visual information must be fulfilled (see ISO - 241-12):

- Clarity
- Discriminability
- Conciseness
- Detectability
- Legibility
- Comprehensibility

General design goals derived from various principles described in [115, 148, 147, 211, 214, 215, 216] [236, 318] are:

- Aesthetically Pleasing:
  - Provide meaningful contrast between screen elements
  - Create Groupings
  - Align screen elements and groups
  - Use color and graphics effectively and simple
- Clarity
- Compatibility with: the user, the task and the product, in other words adopt the user's perspective
- Comprehensibility: Actions, responses, visual representations and information should be in a sensible order.
- Configurability: Permit easy personalization and configuration of settings
- Consistency
- Control: The user must control the interaction
  - Actions result from explicit user requests
  - Action should be performed quickly

  - Actions should be capable of interruption or termination

- Directness: Provide direct means to accomplish the task
- Efficiency: Minimize various movements and actions in order to perform the tasks.
- Familiarity: Concepts and language must be familiar to the end user. Interaction should be natural and in accordance to the user's behavior patterns.
- Flexibility: A system sensitive and efficient to the different needs of different users
- Forgiveness
  - Forgive common and unavoidable human errors
  - Whenever possible prevents errors from occurring
  - Protect against possible catastrophic errors
  - Provide constructive messages when an error occurs
- Predictability: The user must be able to anticipate the needed steps to accomplish the task
- Recovery: The user should be able to *undo* their action and should never loose their work from various errors from their part or system hardware and/or software problems.
- Responsiveness: The system must rapidly respond the the user's requests.
- Simplicity: The interface should be as simple as possible
- Transparency: User should be focused to the task without concern for the mechanisms of the interface

The overall goal of usability can only be reached, if the context of use and the human characteristics are considered in design and evaluation of computer systems.

3.4 Components of User Interface Software - The X Window System case study

User interface software may be divided into various layers: The windowing system, the toolkit and higher level tools. Of course many systems span multiple layers. The windowing system supports the separation of the screen into different regions, usually rectangular.

The X system [267] divides the window functionality into 2 layers: "The window system" which is the functional or programming interface, and the "window manager" which is the user interface. Thus the window system provides procedures that allow the application to draw pictures on the screen and get input from the user while the window manager allows the end user to move windows around and is responsible for displaying the title lines, borders and icons around the windows.

Above the windowing system is the toolkit. The toolkit contains many commonly used widgets such as menus, buttons, scroll bars, text input fields etc. It is a library of widgets where a widget is a way of using a typical input device, such as those mentioned above, to pass a value to the application. Toolkits can be implemented either using or being used by the window system. In the X window system the toolkit is being on top of the windowing system witch result X programmers to be able to use a variety of toolkits, for example xt [159], InterViews [201], Garnet [61], tk [160] and various other toolkits. The xt [159] is a result of designers arguments in agreeing on a single look and feel. So they created an intrinsic layer where different widgets set could be built on top.

A problem that arises by the various toolkits and windowing systems is that people find it difficult to port their code from one toolkit to another. Therefore various virtual toolkits have been developed that hide the differences between the toolkits by providing virtual widgets which later can be mapped to the widgets of each toolkit. There are two type of virtual toolkits those that link to the different actual toolkits on the host machine such as XVT [30] which provides a C or C++ interface that links to the actual Motif, Macintosh, Ms-Windows, toolkits and those that implement the widgets in each style such as Galaxy [19] and OpenInterface [16].

On the top of the toolkit may be higher level tools which help the designer to use the toolkit widgets. In the X window system all communication between the application and the window system use inter-process communication through a network protocol. The following section on tools, successful and unsuccessful tools was taken from [222]

3.5 Tools

User interface software tools helps developers design and implement the user interface. Research on past tools has had enormous impact on today's developers. Virtually all applications today were built using some form of user interface tool. These tools have achieved a high level of sophistication due in part to the homogeneity of today's user interfaces, as well as the hardware and software platforms they run on.

However, conventional GUI (Graphical User Interface) techniques appear to be ill-suited for some of the kinds of interactive platforms now starting to emerge, with ubiquitous computing devices [326] having tiny and large displays, recognition-based user interfaces using speech and gestures, and requirements for other facilities such as end-user programming.

3.5.1 Successful tools

Window Managers and Toolkits: Many research systems in the 1960s, such as NLS [328], demonstrated the use of multiple windows at the same time. Alan Kay proposed the idea of overlapping windows in his 1969 University of Utah Ph.D. thesis [170] and they first appeared in 1974 in his Smalltalk system from Xerox PARC. Many other research and commercial systems picked up the idea from there, notably the Xerox Star, the Apple Macintosh, and Microsoft Windows.

Event Languages: With event languages, the occurrence of each significant event – such as manipulation of an input device by the user – is placed in an event record data structure (often simply called an event). These events are then sent to individual event handlers that contain the code necessary to properly respond to that input. Researchers have investigated this style in a number of systems, including the University of Alberta User Interface Management System [123], Sassafras [266], and others. Event languages have been successful because they map well to the direct manipulation graphical user interface. These systems generate events for each user action with the mouse and keyboard, which are directed to the appropriate application that then must respond. However, the recognition-based user interfaces that are emerging for modalities such as gestures and speech may not map well to this event-based style.

Interactive Graphical Tools: Another important contribution of user interface research has been the creation of what has come to be called interface builders. These are interactive tools that allow interactive components to be placed using a mouse to create windows and dialog boxes. Early research on this class of tools includes Trillium from Xerox PARC [132] and MenuLay from the University of Toronto [69]. The idea was refined by Jean-Marie Hullot while a researcher at INRIA, and Hullot later brought the idea with him to NeXT, which popularized this type of tool with the NeXT Interface Builder. An important reason for the success of interface builders has been that they use graphical means to express graphical concepts (e.g., interface layout). These properties of interface builders can be thought of as providing a low threshold to use, and avoiding a steep learning curve (at least initially). In these systems, simple things can be done in simple ways.

Component Systems: The idea of creating applications by dynamically combining separately written and compiled components was first demonstrated in the Andrew system [241] from Carnegie Mellon University's Information Technology Center. Each component controlled its rectangle of the screen, and other components could be incorporated inside. This idea has been adopted by Microsoft's OLE and ActiveX, Apple's OpenDoc, and Sun's Java Beans [15]. One reason for the success of the component model is that it addresses the important and useful aspect of application building: how to appropriately modularize the software into smaller parts, while still providing significant capabilities and integration to users.

Hypertext: The World-Wide Web (WWW) is another spectacular success of research on user interface software and technology. It is based on the hypertext idea. Ted Nelson coined the term "hypertext" in 1965 and worked on one of the first hypertext systems called the "Hypertext Editing System" at Brown University. The NLS system [105] also had hypertext features. The University of Maryland's Hyperties was the first system where highlighted items in the text could be clicked on to go to other pages [181]. HyperCard from Apple was

significant in helping popularize the idea for a wide audience. For a more complete history of Hypertext, see [230]. Hypertext however did not attain widespread use, however, until the creation of the World-Wide Web system by Berners-Lee in 1990, and the Mosaic browser a few years later.

Object-Oriented Programming: Object-oriented programming and user interface research have a long and intertwined history, starting with Smalltalk's motivation to make it easy to create interactive, graphical programs. C++ became popular when programming graphical user interfaces became widely necessary with Windows 3.1. Object-oriented programming is especially natural for user interface programming since the components of user interfaces (buttons, sliders, etc) are manifested as visible objects with their own state (which corresponds to instance variables) and their own operations (which correspond to methods).

3.5.2 Unsuccessful tools

User Interface Management Systems: In the early 80's, the concept of a *user interface management system* (UIMS) [309] was an important focusing point for the then-forming user interface software community. The term "user interface management system" was coined [167] to suggest an analogy to database management systems. User interface management systems were to abstract the details of input and output devices, providing standard or automatically generated implementations of interfaces, and generally allowing interfaces to be specified at a higher level of abstraction. For every user interface, it is important to control the low-level pragmatics of how the interactions look and feel, which these UIMSs tried to isolate from the designer. Furthermore, the standardization of the user interface elements in the late 1980's on the desktop paradigm made the need for abstractions from the input devices mostly unnecessary. Thus, UIMSs fell victim to the moving target problem.

Formal Language Based Tools: A number of the early approaches to building a UIMS used techniques borrowed from formal languages or compilers. For example, many systems were based on state transition diagrams (e.g.,[154, 229, 325]) and parsers for context free grammars (e.g., [235]). Initially these approaches looked very promising. However, these techniques did not catch on for several important reasons that can serve as important lessons for future tools. The direct manipulation style of interface [294] was quickly coming to prominence. In direct manipulation interfaces, the role of dialog management is greatly reduced. In these systems it is very easy to express sequencing (and hard to express unordered operations). As a result, they tend to lead the programmer to create interfaces with rigid sequences of required actions. Another reason that some systems did not catch on is that had a high threshold for using them because they required programmers to learn a new special purpose programming language (in addition to their primary implementation language such as Pascal, C, or C++).

Constraints: Many research systems have explored the use of constraints, which are relationships that are declared once and then maintained automatically by the system, for implementing several different aspects of a user interface. Examples include Sketchpad [303, 303], ThingLab [59], HIGGENS [144], CONSTRAINT [316], Garnet [61], Rendezvous [138], Amulet [223], and subArctic[145].

Constraint systems offer simple, declarative specifications for a capability useful in implementing several different aspects of an interface. However, constraint systems have yet to be widely adopted beyond research systems. One of the central reasons for this is that programmers do not like that constraint solvers are sometimes unpredictable. If there is a bug in a constraint method, it can be difficult to find. Furthermore, the declarative nature of constraints is often difficult to master for people used to programming in imperative languages. It requires them to think differently about their problems, which also contributes to having a high threshold.

One area of user interface design for which constraints do seem successful is the layout of graphical elements. Systems such as NeXTStep and Galaxy [19] provided a limited form of constraints using the metaphor of "springs and struts" (for stretchy or rigid constraints) which could be used to control layout. These and other metaphors have found wider acceptance because they provided a limited form of constraints in a way that was easier to learn and more predictable for programmers.

3.6 Information Exploration and Visualization

3.6.1 Information Exploration

Information exploration should be a joyous experience, but many commentators talk of information overload and anxiety. Exploring information collections becomes increasingly difficult as the volume and diversity grows.

Computers are a powerfull tool for searching, but traditional user interfaces are a hurdle for novice users and an inadequate tool for experts.

Interfaces for searching structured databases and textual-document libraries are good and getting better, but searching multimedia document libraries is still in primitive stage. Current approaches to locating images, sound, videos or animations depend on a parallel database or document search to locate the items.

Recent advances in computer algorithms may enable greater flexibility in locating items in multimedia libraries. Most systems nowadays are moving towards graphical specification of query components such as:

- *Photo Search*: Finding photos with images is a substantial challenge for image-analysis researchers, who describe this task as *query by image content* (QBIC).
- *Map search*: Search by features is becoming possible because the tools used to built maps preserve the structural aspects and the multiple layers in maps.
- *Design or Diagram search*: Diagramming tools for making flowcharts or organization charts can add search capabilities.
- *Sound search*: Searching sounds with the use of sound as query input is also a substantial challenge for researchers. Finding a spoken word or phrase in database of telephone conversations is still difficult, but is becoming possible, even on a speaker independent basis.
- *Video search*: Searching video is also a big challenge to the research community which involves more than simply searching each of the frames.
- *Animation search*: Animation-authoring tools are still in early stages of development, but it might be possible to specify searches for certain kinds of animation.

3.6.2 Information Visualization

As computer speeds and displays resolution increases, information visualization and graphic interfaces are likely to have an expanding role. Overall, the bandwidth of information presentation is potentially higher in the visual domain than it is for media reaching any of the other senses. Users can scan, recognize, and recall images rapidly, and can detect subtle changes in size, color, shape, movement, or texture. So, as visual approaches are explored appealing new opportunities are emerging.

Information visualization is a demanding and important task. Trying to characterize the multiple information-visualization innovations and sort out the numerous prototypes Shneiderman [293] concluded on the *data type by task taxonomy (TTT)* schema of information visualization. This schema is divided in two parts the *data types* and the *tasks*. Seven major data types and seven tasks were outlined which are:

Types

- *1-D Linear Data*: An early approach to dealing with one dimensional-data sets was the *bifocal display* [299]. Another attempt was using fixes size space with a scrollbar like display called *value pairs* [82]. Other attempt were SeeSoft [18], or lines in Hamlet, Document Lens [268], information mural algorithms [158].
- *2-D Map Data*: Examples include geographical-information systems, which are a large research and commercial domain [189, 101]. Information visualization researchers have used spatial displays of document collections [180, 329] organized proximally by term co-occurrences.
- *3-D World*: Examples of three-dimensional computer graphics design are numerous, but information-visualization work in three-dimensions is still novel. A three-dimensional desktop is thought to be appealing to the user, but disorientation, navigation, and hidden data problems remain [74].
- *Temporal Data*: Time lines are widely used. The distinction of *temporal data* are that items have a start and finish time and that items may overlap. There are many tools for visualization of time such as perspective wall [269], Life-Lines [254]. Temporal data visualizations also appear in systems for editing video data, composing music, or preparing animations.

- *Multidimensional Data*: Most relational and statistical database contents are manipulated as multidimensional data in which items with n attributes become points in an *n-dimensional space*. It can be represented using a three-dimensional scattergun, but disorientation and occlusion can be problems. Various attempt in visualizing Multi-Dimensional data are: [84, 121, 108, 85, 174, 265, 313]
- *Tree Data*: Tree structures are collections of items where each item (besides the root) has a link to one parent item. Items and the links between them have multiple attributes. Interface representation of tree data can use the style of indented labels used in table of contents, a node-link diagram or a treemap. The treemap approach was successfully applied to to libraries, computer directories, sales data, business decision making [42] and web browsing [218, 220].
- *Network Data*: Network visualization is an old but still imperfect art because of the complexity of relationships and user tasks. Commercial packages can handle small networks such as Netmap. New interest in this topic has been spawned by attempts to visualize the World Wide web [39, 133].

Tasks

- *Overview*: Gain an overview of the entire collection
- *Zoom*: Zoom in on items of interest
- *Filter*: Filter out uninteresting items
- *Details-on-Demand*: Select an item and group and get details when needed
- *Relate*: View relationships among items
- *History*: Keep a History in actions to support undo, replay, and progressive refinement
- *Extract*: Allow extraction of subcollections and of the query parameters

As in the case of search, users are assumed to be viewing collections of items, where items have multiple attributes. In all seven data types the items have one or more attributes. The data types of the TTT characterize the task domain information objects and are organized by the problems that users are trying to solve. These seven data types represent an abstraction of the reality and many variations of them can be met.

4 Speech and natural language interfaces

Speech is the most natural form of communication between humans, but it has several limitations when used in interaction with computers. Although speech recognition technology has been studied actively during the past decades and highly sophisticated recognizers have been constructed, it is considered as the main obstacle in speech systems. Unrealistic assumptions and improper application design is the main reason, but with appropriate interaction techniques most limitations can be overcome and successful applications can be constructed.

The main problem is the variations in speech signal (context, style, speaker and environmental variability) but some sources of variations can be controlled with the use of speaker and environmental adapted recognition grammars and acoustic models. Also, spontaneous spoken language contains a lot of ungrammatical elements such as hesitations, false starts, repairs and overlapping which cause complexity and make speech harder to understand. Finally, people are used to talk differently to computers than to other people by often altering their speaking styles.

In the first part of this section we briefly compare speech and GUI interfaces, highlighting strengths and weaknesses of speech as an input/output modality. Next we review the most common kinds of speech applications and analyze how the capabilities and features of a speech recognition system can affect the interaction of a speech application. We delve more into the component technologies used in speech applications, namely speech recognition and synthesis and language processing and understanding in section 4.3. Finally section 4.4 reviews speech tools and technologies for development of speech applications.

4.1 Speech as an Input/Output Modality

With GUIs everything the user wants to do at any given time must be presented in the screen. Although this is considered to be a limitation for GUIs, in the case of speech interfaces the lack of visual information requires users to memorize all meaningful information. The temporal nature of speech loads the short-term memory and takes up the linguistic channel, which makes speech interfaces unsuitable for some tasks. Also, users interacting with speech interfaces don't have the same feeling of control usually offered by GUI interfaces.

Spoken interaction may be faster if users immediately can say what they want to achieve without going through menu hierarchies. Spoken messages may also be more expressive and convey richer information compared to GUI actions, such as the selection of similar objects among a large number of them. However the freedom and efficiency that speech gives to user makes speech harder for the computer to handle. It is also hard for users to know the limitations of what they can say and how to explore the set of possible tasks they can perform.

As an output channel, speech is too slow because of its sequential nature while GUIs convey information in parallel thus making them suitable for presenting a large amount of information to the user.

4.2 Speech applications

The first speech applications were telephone-based interactive voice response systems, which used speech outputs and telephone keys for interaction. Such applications (possibly best examples of widely commercial speech applications) are designed to replace human operators. Besides IVR, other telephony applications have dominated the field such as information services(timetable, weather forecasting and e-banking), e-mail applications and voice portals. These systems are fairly sophisticated and include state-of-the art recognizers, natural language understanding and response generation components but still integration is the key issue for successful applications ([213]).

Desktop applications such as dictation systems and command and control applications have also been shown. Dictation systems are relatively popular to special user groups. Command and control applications usually control existing graphical applications, without using mouse/keyboard, which can be very useful for devices such as PDAs.

4.2.1 How speech recognizer features affect speech applications

The capabilities and features of a speech recognition system can affect the design and interaction of a speech application. As shown in 4.2.1 ([311]), several features affecting a speech recognizer's suitability for different

tasks is depicted. Most difficult and desired properties are shown in the far right column.

Vocabulary size and recognition grammars characterize the interaction possibly better than other properties. For example, it is possible to construct a speech-only e-mail application with a dozen of words, but for building an information retrieval system, at least a middle-sized vocabulary is needed. The possibility to change or dynamically construct grammars also affects interaction: for example allows the system to be context-sensitive and use user profiles with personalized recognition grammars.

Communication style can vary from speaker-dependent, discrete speech and half-duplex to speaker-independent, continuous and full duplex. Speaker-dependent or adaptive models are suitable for some applications (e.g. dictation) while speaker-independent in public application (information services). Although with current recognizers there is no need to speak in a discrete manner, it usually helps if words are pronounced clearly and properly. Spontaneous speech is still challenging and world-spotting is used. Finally, capabilities like barge-in which can be used to interrupt system output can influence the design (the system can generate longer and more informative responses).

Usage conditions can vary from clean to hostile environments and low (public mobile phone usage) to high quality channels(close-talk microphones). Even with state-of-art recognizers, performance can dramatically suffer is usage conditions don't match recognizer training ones. This is usually compensated by usage of different acoustic models for different conditions.

	Vocabulary and language		
Vocabulary size	small	middle-size	very large
Grammar (LM)	phrases	CFG	n-gram
Extensibility	fixed	changeable	dynamic
	Communication style		
Speaker	dependent	adaptive	independent
Speaking style	discrete	continuous	spontaneous
Overlap	half-duplex	barge-in	full-duplex
	Usage conditions		
Environment	clean	normal	hostile
Channel quality	low-quality	normal-quality	high-quality

Table 1: Properties of speech recognition systems

4.3 Speech application component technologies

Speech recognition and speech synthesis technology is the basis of spoken dialogue and multimodal dialogue systems. Since the analytical study of these technologies is more than the brief review given here the interested reader should check [264].

4.3.1 Speech recognition technology

Speech recognition is the key component technology. For a brief review see [332].

The recognition process

Automatic speech recognition (ASR), is the process of transforming the digitized acoustic signal into words. When speech is produced as a sequence of words, *language models* or artificial grammars are used in order to constrain the combination of words. General language models approximating natural language are specified in terms of a *context-sensitive grammar*. One measure of the difficulty of the task, combining the vocabulary size and the language model is *perplexity*, loosely defined as the geometric mean of the number of words that can

follow a word after the language model has been applied. In order to recognize an input sentence the following steps must be followed: The audio signal is digitized and is transformed into a set of useful measurements or features (*feature extraction*) at a fixed rate, typically every 10-20 msec. For more information on *feature extraction* technology see [202, 136, 113, 41], [300, 135, 139, 165, 194, 146]. These measurements are then used to search for the most possible word candidates through the constraints imposed by the acoustic, lexical and language model. For this process training data are used to determine the values of the model's parameters. The dominant recognition approaches used the past fifteen years are *Hidden Markov Models* (HMMs). For more information on language models and HMMS see [143, 55, 50, 263, 94].

Recognition performance

Recent advances of automatic speech recognition (ASR) technology have achieved good word accuracy in adverse conditions (high background noise or telephone speech) for specific tasks (small to medium vocabulary size). The past decade significant progress has been made in speech recognition technology. According to [272] word error rates continue to drop by a factor of two every two years. The best system in 1994 achieved an error rate of 7.2% on read sentences drawn from North American business news [242]. In the Air Travel Information Service (ATIS) domain, word error rates of less than 3% has been reported for a vocabulary of nearly 2,000 words and a bigram language model with a perplexity of around 15. In clean conditions, high accuracy can be achieved for large vocabulary tasks, e.g., dictation. After working on isolated-word and speaker-dependent systems for many years, since 1992 the community has moved towards very-large-vocabulary (20,000 words and more), high-perplexity (200 and more) speaker-independent, continuous speech recognition.

Speech recognition systems have become much more robust in recent years with respect to both speaker and acoustical variability. In addition to achieving speaker-independence, many current systems can also automatically compensate for modest amounts of acoustical degradation caused by the effects of unknown noise and unknown linear filtering [32, 195, 114]. Substantial progress has also been made over the last decade in the dynamic adaptation of speech recognition systems to new speakers, with techniques that modify or warp the system's phonetic representations to reflect the acoustical characteristics of individual speakers [116, 142, 283].

4.3.2 Speech synthesis technology

Speech synthesis research predates other forms of speech technology by many years. In the early days of synthesis, research efforts were devoted mainly to simulating human speech production mechanisms, using basic articulatory models based on electro-acoustic theories. Though, this modeling is still one of the ultimate goals of synthesis research. Advances in computer science have widened the research field to include Text-to-Speech (TTS) processing in which not only human speech generation but also text processing is modeled [35]. Because this modeling is done by applying a set of rules e.g., from phonetic theories and acoustic analysis, the technology is referred to as *speech synthesis by rule*. In contrast to this traditional rule-based approach, a corpus-based approach has also been pursued. In the corpus-based work, well-defined speech data sets have been annotated at various levels with information, such as acoustic-phonetic labels and syntactic bracketing. In TTS systems, speech units that are typically smaller than words are used to synthesize speech from arbitrary input text. To bring objective techniques into the generation of appropriate speech units, *unit-selection synthesis* has been proposed [224, 234, 275], [307]. In unit-selection synthesis, speech units are algorithmically extracted from a phonetically transcribed speech data set, using objective measures based on acoustic and phonetic criteria.

The main drawback of synthesized speech is that it doesn't sound natural. For synthesis of natural-sounding speech, it is essential to control *prosody*, to ensure appropriate rhythm, tempo, accent, intonation and stress. Segmental duration control is needed to model temporal characteristics, just as fundamental frequency control is needed for tonal characteristics ([45, 71, 75, 175]).

The field of *spoken language generation* is in its infancy, with very few researchers working on systems that deal with all aspects of producing spoken language responses, i.e., determining what to say, how to say it, and how to pronounce it. Within speech synthesis, research on controlling intonation to signal meaning and discourse structure is relevant to the problem.

4.3.3 Semantics grammars and understanding

The semantics

We understand larger textual units by combining our understanding of smaller ones. The main aim of linguistic theory is to show how these larger units of meaning arise out of the combination of the smaller ones. This is modeled by means of a grammar. *Computational linguistics* tries to implement this process in an efficient way. It is traditional to subdivide the task into *syntax* and *semantics*, where *syntax* describes how the different formal elements of a textual unit, most often the sentence, can be combined and *semantics* describes how the interpretation is calculated. The *grammar* consists of a lexicon, and rules that syntactically and semantically combine words and phrases into larger phrases and sentences. In current research, very simple grammar models are employed, e.g., different kinds of *finite-state grammars* that support highly efficient processing. Some approaches do away with grammars altogether and use statistical methods to find basic linguistic patterns.

A very advanced and wide-spread class of linguistic formalisms are the so-called *constraint-based grammar* formalisms which are also often subsumed under the term *unification grammars*. Among the most used, constraint-based grammar models are: Functional Unification Grammar (FUG), [172] Head-Driven Phrase-Structure Grammar (HPSG),[255] Lexical Functional Grammar (LFG) [64], Categorial Unification Grammar (CUG) [129], [166] and Tree Adjunction Grammar (TAG) [164]. For these or similar grammar models, powerful formalisms have been designed and implemented that are usually employed for both grammar development and linguistic processing, e.g, LFG [64], PATR [292], ALE [76], STUF [60] ALEP [36],[37] TDL [182],TFS [104].

However disambiguation using knowledge-based techniques requires the specification of too much detailed semantic information to yield a robust domain-independent parser. The availability of large bodies of machine readable textual data has provided impetus to statistical approaches to disambiguation. Some approaches use stochastic language modeling inspired by the success of HMM-based lexical category disambiguation. For example, probabilities for a probabilistic version of a context-free grammar (PCFG) can be re-estimated from treebanks or plain text [112]. More recently, attempts have been made to use statistical induction to learn the correct grammar for a given corpus of data, using generalizations of HMM maximum-likelihood re-estimation techniques to PCFGs ([188]). [250] and [279] showed that constraining the analysis considered during re-estimation to those consistent with manual parses of a treebank, reduces computational complexity and leads to a useful grammar.

A semantic description of a language is a nicely stated mechanism that allows us to say, for each sentence of the language, what its truth conditions are. Just as for grammatical description, a semantic theory will characterize complex and novel sentences on the basis of their constituents: their meanings, and the manner in which they are put together. Almost all current large scale implementations of systems with a semantic component are inspired to a greater or lesser extent by the work of Montague [219],see [46, 34, 37]. Currently, the most pressing needs for semantic theory are nowdays these of achieving wider and more robust coverage of real data.

Spoken Language Understanding

Spoken Language Understanding involves two primary technologies: *Speech Recognition* and *Natural Language Understanding*. The integration of these two technologies has great advantages. *Language Understanding* can supplement *speech recognition* with informations that are not clearly represented in text such as syntax, semantics and pragmatics. With *Natural Language* analysis, based predominantly on complete parsing of grammatically correct sentences *(written text)*, traditional *Natural Language* analysis often do very poorly when faced with transcribed spontaneous speech. To combat the mismatch between existing SR and NL modules, two trends have been observed. The first is an increased use of *semantic*, as opposed to *syntactic grammars*. Such grammars rely on finding an interpretation without requiring grammatical input. Because *semantic grammars* focus on meaning in terms of the particular application, they can be more robust to grammatical derivations. The second observed trend is the *n-best interface*. In this approach the connection between the speech recognizer and the natural language is completely serial. In this approach recognizer passes to the natural language the *n-best* (n-most possible) transcriptions and the NL calculates scores on these, with the use of grammatical and other knowledge sources.

With few exceptions *Natural Language* research has focused on the input side *Understanding on spoken*

input. The use of output technologies is an important challenge to spoken language systems.In particular reliable technique are needed in order to:

- decide when it is appropriate to provide a spoken output in conjunction with some other output and/or to instigate a clarification dialogue in order to recover from a potential misunderstanding
- generate the content of spoken output given the data representation, context and dialogue state, and coordinate it with other outputs when present
- synthesize a natural, easily interpreted and appropriate spoken version of the response taking advantage of the context and dialogue state and emphasize on certain information or to express urgency,
- coordinate spoken outputs to guide the user toward usage better adapted to system capabilities.

By coordinating inputs and outputs, the system can guide the user toward better adaptation to the particular system.

4.4 Tools and standards

4.4.1 VoiceXML

The VoiceXML Forum (an organization founded by Motorola, IBM, AT&T, and Lucent to promote voice-based development) [21] introduced a new language called *VoiceXML* based on the legacy of languages already promoted by these four companies. In March 2000 version 1.0 was released and in October 2001 the first working draft of the latest VoiceXML 2.0 was published as a W3C recommendation [20]. VoiceXML has the potential to boost the development of voice-based applications much like just as HTML did for the development of web-based applications. It's major advantage, is the ability to provide web content using only voice as an input modality, making it accessible from devices like phones - fixed or mobile - thus, reaching a much greater audience.

VoiceXML browsers consists of an interpreter and a set of VoiceXML documents. VXML supports dialogues in the form of menus and forms, sub-dialogues and embedded grammars. The *voice browser* renders the VoiceXML documents as a sequence of the two-way interaction between the system and the end user, by using the core VoiceXML interpreter, and software components such as *Automatic-Speech-Recognition*(ASR) and *Text-To-Speech*(TTS).

Many companies provide solutions (by introducing custom tags of objects) to implement and test complete VoiceXML-based applications and *voice portals* like the Nuance Voice Platform ([1]), which provides an easy to use complete development environment. Other commercial offerings include servers for deploying these applications ([2]), voice browsers, VoiceXML editors, grammar development tools and more ([3, 4, 5]). There are also open source VoiceXML tools, such as Canregie Mellon's OpenVXI interpreter [6].

5 Spoken Dialogue Systems Design

In this section, we review efforts in defining design principles and creating tools for building multimodal dialog systems with emphasis on the speech modality. General design principles for architecting and building such systems are reviewed and challenges are outlined. The focus is on system architecture, application and speech interface design, data collection and evaluation tools. We conclude that modularity, flexibility, customizability, domain-independence and automatic dialog generation are some important features of successful dialog systems and design tools.

5.1 Introduction

Building a successful dialog system requires inter-disciplinary expertise that goes beyond having state-of-the-art speech and natural language processing technology. The modules of a dialog system include: *automatic speech recognition* (ASR), *text-to-speech* (TTS) *synthesis*, *natural language understanding* (NLU), *application manager*, *dialog manager*, *database*, *controller/event handler*, and, for multimodal systems, *graphical user interface* (GUI), *gesture/sign recognition*, *visual speech recognition*. Due to the lack of space we will not describe in detail the building blocks of a dialog system. Instead we will focus on the system design process, provide guidelines and outline challenges. Dialog system design is typically comprised of four steps: (i) architectural design, (ii) application design and data collection, (iii) speech and natural language interface design and (iv) user feedback and evaluation. Although these steps can be carried out more all less independently of each other, some degree of coordination is needed to guarantee consistency and smooth integration.

The process of dialog system and interface design is an iterative one, as is the case for applications with traditional input and output modalities. The main reason for the need of iterative enhancement is the lack of accurate user models that could provide objective measures of system performance. In addition, application-specific data needs to be collected and labeled. Given the iterative nature of system design it is important to provide guidelines and tools that can reduce the concept-to-prototype development time and the number of iterations required. In addition, this set of tools has to be *flexible* and *general* enough to be able to build successful, *modular* and *upgradeable* dialog systems, with *reusable* components.

Natural language dialog system design has been an active research field for over four decades. Early theoretical contributions to the field were made by scientists from the areas of artificial intelligence and linguistics, e.g., [297]. The advent of robust speech recognition technology and increased computing power towards the end of the 80's made applications in the area of spoken dialog systems possible. As a result, speech engineers started applying machine learning techniques that have been shown to be successful for speech recognition, to natural language understanding and dialog management (see [92] for historical notes). The travel reservation domain and specifically the DARPA Airline Travel and Information System (ATIS) task was the driving force in the beginning of the 90's and provided the seed for most prototype dialog systems in industry and academia [253, 287, 185].

In terms of system architecture, dialog systems have evolved from monolithic, hard-coded system design to a *modular* architecture of re-usable and programmable building blocks [253, 287, 288, 323, 225]. *Stateless* servers were introduced in [253, 287], where dialog and application state information is kept in a template that is passed between and updated by the various modules. The communication between application and interface modules was standardized in [247] using the *Voice Interface Language* (VIL). Concepts such as *abstraction*, *inheritance*, *encapsulation* and *polymorphism* were borrowed from computer science for semantic and dialog state representation [323]. Novel ideas for application and interface design such as *customizability* and *automatic dialog generation* were introduced in [245, 246, 184]. Finally multimodal input/multimodal output system design principles and an interface with personality was demonstrated in [225]. Spoken dialog technology has been successfully used for building telephony applications (there speech provides clear advantages over the traditional touch-tone user interface). A platform architecture that supports multiple telephony channels and applications was proposed in [338]. Systems continue to evolve and new design principles are introduced that improve functionality, modularity and versatility.

The organization of this section is as follows: We describe the steps involved in dialog system design and provide guidelines for building such systems, in terms of architecture, data collection, application and dialog design, speech and natural language interface, and evaluation. We then review the set of tools available for

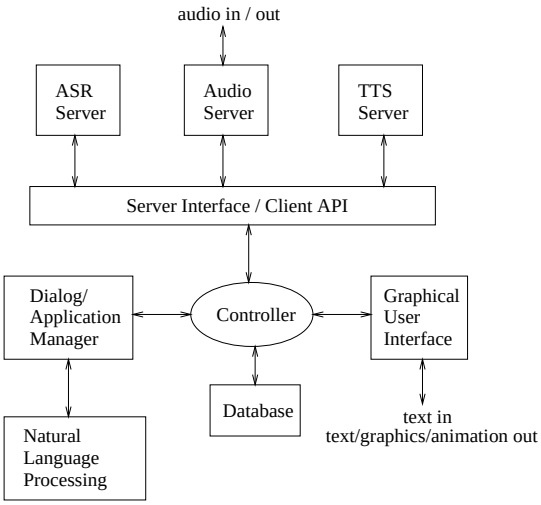

Figure 1: Multimodal system architecture.

building dialog systems and conclude the paper.

5.2 Design Guidelines

A fundamental issue in designing dialog systems is the choice of input and output modes in the user interface. Few guidelines exist for selecting the appropriate mix of modalities [53], however, it is clear that a spoken language interface is not the best choice for all applications. Other important issues include selecting a modular architecture and a *cooperative* interface. Some of these issues are discussed in the following sections. Specifically, the features of a successful system and the challenges that lie ahead are outlined. Note that general principles from computer science such as abstraction, inheritance and encapsulation should also be applied to software development of spoken dialog systems.

5.2.1 Architectural Guidelines

The architecture of dialog and multimodal systems has steadily evolved from a monolithic to a *modular* architecture with well-defined communication protocols between modules, e.g., [40]. In Fig. 1, a typical architecture is shown for a multimodal system with emphasis on the speech modality (adapted from [247]). The audio, speech recognizer and text-to-speech servers are controlled by a client layer through a well-defined *application programming interface* (API). The client brokers connections between the audio and ASR server (speech input), and the TTS and audio server (speech output). Multiple ASR, TTS or audio servers can be handled by the client API. The controller determines the generic multimodal application control flow, merges the input streams (speech, text, gestures) and synchronizes the output streams (speech, text, graphics, animation) [225].

The typical control flow is as follows: get transcription(s) from ASR and/or GUI, send them to the dialog manager/NLU module for processing, get results from dialog manager and present them to the user. Database information can be optionally requested from the controller by the application manager. A scripting language is often used for the controller, e.g., PERL in [246, 225]; a new scripting language is defined in [287]. Variations on the architecture in Fig. 1 can be found in the literature, e.g., in [287] some of the functionality of the application manager has been transferred to the controller for increased modularity; this architecture was adopted by the DARPA Communicator project.

There is no consensus among researchers on the communication protocol between the various modules. Typically both synchronous and asynchronous (event-driven) communication are allowed, although synchronous communication is preferred when possible. In some systems, information is communicated among and within servers through message-passing. The message contains all state information and thus the servers are *stateless* [253, 287, 225]. However, the format and content of these messages are very different among systems. The APIs

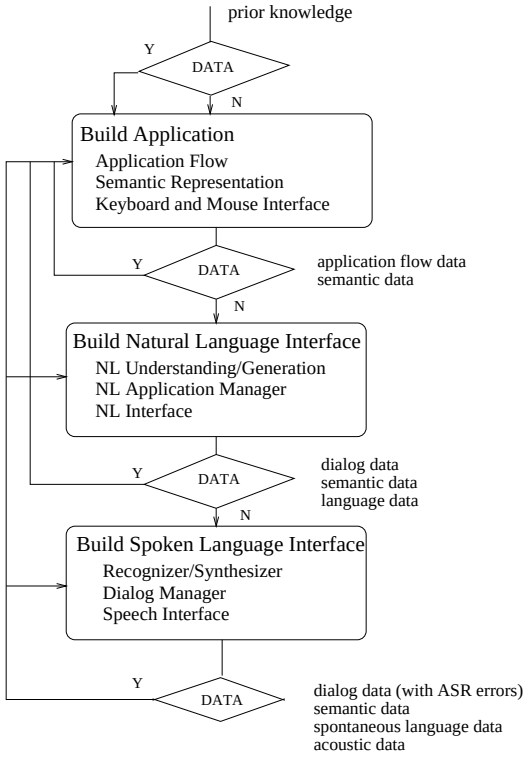

Figure 2: Iterative dialog system design using (optional) multi-stage data collection.

for the communication between the dialog manager, controller, NLP and database modules need to be carefully defined. In [247], the communication between the application manager and the audio, recognition and synthesis server is done through the Voice Interface Language (VIL).

5.2.2 Data Collection and WoZ Experiments

Data collection and analysis form a crucial part of system development. In Fig. 2, we show the various stages of application development, as the interface evolves from keyboard and mouse to spoken language. Data can be collected at various stages either automatically or assisted by supervisors (wizards). The types of data collected are acoustic, lexical, semantic and application/dialog flow data as shown in the figure. Word level transcriptions and application-specific semantic labels are assigned to the data either manually or semi-automatically (using the output of the recognition and understanding module). Collected data are typically very much dependent on the system configuration and the application scenario. As a result, dialog system development is an iterative process: collected data are used to improve the system, and more data are collected using the updated system configuration.

Wizard of Oz (WoZ) scenarios have been used extensively for interface design [99]. WoZ systems can be categorized into: (i) *fully wizarded* systems, (ii) *wizard-supervised* systems. Fully wizarded dialog systems use two wizards: one transcribing the user input and another typing the system output. As a result, the response time is large and the data obtained can be unrealistic. In practice, WoZ scenarios use a fully automatic prototype system and the wizard acts as a supervisor for the system, correcting errors and intervening when things break down, e.g., [140]. WoZ scenarios are especially useful for tactile input systems to emulate the voice modality ("voicing-over" of an application) [257].

5.2.3 Application and dialog manager design

Designing a good application is the first step towards building a successful dialog system. The importance of application design is often overlooked by the speech community, where the emphasis is on a unimodal speech interface. As a result, applications often focus on the interface, and have limited functionality. In Fig. 2, an *application-centric* approach to dialog system design is outlined, where the application evolves as the interface is augmented with natural language and spoken language capabilities. The main advantages of this approach are: increased modularity (separation of interface and application), flexible multi-stage data collection as the interface and the application evolves, and an interface that can easily be tailored for "universal access", i.e., create the appropriate mix of modalities based on the way the application is accessed (telephone, personal digital assistant or PC).

One important application area of dialog systems is the *personal assistant/agent*, which encompasses a variety of sub-applications such as name-dialing, messaging (voice, e-mail, fax), information retrieval (news, weather), finance (banking, stock trading), travel reservations (air, car, hotel) etc. In [245, 246], a *hierarchical top-down* approach to application design is proposed. The services that the agent supports are organized in a hierarchical tree-structure which users navigate using voice. A similar hierarchical organization is used for navigating the world wide web using a mouse interface in Internet portals such as Yahoo and Excite.

For large dialog applications, the manual annotation of all possible state transitions can be a gruesome task. In [245, 246], a paradigm for *automatic application and dialog flow generation* is proposed. Specifically, given a hierarchical tree-structure of dialog states the dialog flow is generated automatically as follows: the system can transition to states that are children nodes, sibling nodes or the parent node of the current application state. Automatic dialog generation can also be based on user profile information and database constraints. Automatic and *dynamic* dialog flow generation is a challenging new direction in dialog management.

An important feature of any good application is *customizability*. This can be achieved by building a user profile either by explicitly querying the user or implicitly based on past interaction between the user and the system. In [184], the user profile determines the services to be included in a particular application as well as the format of the data presented to the user. For example, in an information retrieval application the user can specify the genre of news or sports reports he is interesting in and the format the news headlines are presented in. The importance of tailoring application content and form to the user's needs is clearly demonstrated by web-based news and information retrieval gateways such as Pointcast.

5.2.4 Speech and natural language interface design

Spoken and natural language interface design is the most challenging aspect of building a dialog system. A variety of open research problems and practical issues have to be tackled successfully and expertise from different disciplines has to be combined. Clearly good recognition and understanding performance are vital. In addition, the dialog system should allow for user exploration, provide alternate routes to completing a task, and should never let the user get lost or trapped. *Versatility*, *customizability*, *cooperation* and *supervision* (on a need-only basis) are some of the features of a good interface. Active research issues in dialog system design relevant to the goals outlined above include:

- handling spontaneous speech phenomena and robust speech recognition,
- dynamic creation of recognition grammars based on dialog history,
- dynamic generation of system prompts based on updated user beliefs,
- dynamic control of user/system initiative of the dialog as a function of user progress.

Some practical issues that are equally important are : echo cancellation of system prompts, timing in turn-taking (system time-outs), user or system interruption of dialog flow (barge-in) etc. Prototypes of dialog systems are starting to incorporate features that improve the flexibility of the spoken language interface, e.g., automatic relaxation of query constraints for queries with no matches, two-levels of system supervision, implicit confirmation of user input, shorter prompts, optional spelling of proper nouns [185]. Even more ambitious goals are to provide the interface with *intelligence* and *personality* [225, 306].

A *user-centric* approach to dialog system design can help address some of the issues raised above. A static or dynamic user model can be used to predict user intent given the current state of dialog. Grammar fragments can be associated with each of the user's intentions and a dialog state dependent grammar can be automatically generated as a weighted combination of these grammar fragments. In addition, a model of user beliefs can be used during generation of system responses; the user belief model is updated based on user input and system responses. User-centric grammar design and dialog management can help customize and automate the dialog system design process.

As mentioned above, dialog system design goes beyond building state of the art components, e.g., speech recognizer, natural language understanding, dialog manager. An important focal point of dialog system design is the *interaction* between the different modules. For example, acoustic confidence scores computed in the recognizer can be used to improve the performance of the understanding module and to determine the appropriate dialog strategy in the dialog manager. Dialog history can be used to improve recognition and understanding performance [258]. How to integrate the various sources of information to improve performance and enhance user experience remains an open research issue.

Finally designing generic modules that are *application independent* to the degree possible (clearly some application-dependent semantic representation is needed for designing the understanding module) is the ultimate goal of dialog system design. Universal algorithms and tools that can be applied with minimal modifications across applications will significantly speed up development time and create new challenging application areas.

5.2.5 User Feedback and System Evaluation

User feedback is a very important source of information for the designer of dialog systems. Despite recent progress, user interface and application design is mostly an empirical field [99]. Focus groups and user-experience forums can be used as a "proof of concept", i.e., to determine if the application and interface meets user's needs and preferences.

Little formal work can be found in the area of dialog system evaluation. Objective evaluation metrics exist for some of the modules of a dialog system, e.g., word and semantic label accuracy for the recognizer and understanding module, respectively, provided that transcribed and semantically annotated text corpora are available. End-to-end objective measures such as task completion and total number of dialog turns are often used to measure the functionality and efficiency of dialog systems.

Clearly, the ultimate judge of system performance is the user.Subjective measures, such as system satisfaction, perceived efficiency, flexibility and robustness of the system can provide valuable metrics for the evaluation and iterative enhancement of a system. In [140], a mix of subjective and objective criteria was used to evaluate the ATIS systems. In [322], the correlation between objective (word accuracy, task completion, number of dialog turns) and subjective criteria (user satisfaction) was computed; the hope is that the relationship between objective criteria and user-satisfaction is more or less task-independent.

5.3 Tools

An important effort to provide a set of tools for spoken dialog system design was made at the CSLU at Oregon's Graduate Institute. The toolbox provides the modules needed for system building and a graphical user interface for application design. It has been successfully used by non-experts for building simple applications. The toolbox can be downloaded from the web; for a review see [305].

Most commercially available tools for dialog system design have limited natural language understanding capabilities, e.g., [314]. Dialog flow is typically modeled with a finite state machine and each dialog state corresponds to a single application action. In [314], tools for user-centric grammar design are provided, i.e., dialog state grammars are the union of all grammars that correspond to actions that the user can request. In general, current tools require a significant amount of application-specific work, e.g., manual writing of semantic grammars [287]. Further, most of the tools are designed with a specific set of applications in mind (communication assistant, travel reservation) and tuning of the modules is required for each new application domain.

A set of dialog tools should be able to automatically design an application based on transcribed and semantically annotated (in an action/attribute notation) data. In addition, generic dialog strategies should be available for groups of applications, e.g., database queries, information retrieval and browsing, so that the dialog

and application flow are generated automatically based on a user profile. More research is needed to advance the state of the art and achieve these goals.

5.4 Relation to WP10

We have reviewed some important principles and trends in dialog system design. Previously dialog system development has relied on collecting many application-specific examples. Dialog systems were usually not extendable to other applications domains and provided little or no customizability to the user. Our work is motivated by the desire to learn how the various building blocks of a dialog system interact and how a dialog session can be automatically and dynamically generated based on user's preferences. The ultimate goal is to provide flexible and upgradable dialog systems, with general and reusable components. Designing an interface that is versatile and customizable is essential for enhancing user experience. More research is needed to meet these challenges. Modular architecture, automatic dialog generation, user modeling and the voice interface language are important contributions towards achieving these goals.

6 Multimodal Interfaces

As defined in [240], multimodal interfaces process two or more combined user input modalities such as speech, pen, touch, manual gestures, gaze and head and body movements in a coordinated manner with multimedia system output. This is a paradigm shift away from conventional WIMP interfaces towards more flexible, efficient and powerfully expressive means of human computer interaction. Multimodal interfaces are expected to be easier to learn and use, more robust and more adaptable to the user, tasks and usage environment.

In this section, we motivate the use of multimodal interfaces and briefly review some examples of multimodal applications. Then we examine topics such as multimodal interaction (mainly fusion and integration techniques) and dialogue management handling. Finally, we examine issues regarding to system architectures and conclude with various efforts for standardizing multimodal interaction specially on the web.

6.1 Motivation

As shown in [86], multimodal interfaces may have many advantages: they prevent errors, bring robustness to the interface, help the user to correct errors or recover from them easier, bring more bandwidth to the communication and add alternative communication methods to different situations and environments. Disambiguation of error-prone modalities using multimodal interfaces is the main motivation for the use of multiple modalities in many systems. As shown in [238] error-prone technologies can compensate each other, rather than bring redundancy to the interface and reduce the need for error correction.

It should be noted, however, that multiple modalities alone do not bring these benefits to the interface: currently there is too much hype in multimodal systems, and the use of multiple modalities may be ineffective or even disadvantageous. Oviatt [239] has presented common misconceptions (myths) of multimodal interfaces most of them related to the use of speech as an input modality.

6.2 Example applications

From the historical perspective, multimodality offers promising opportunities, as presented Bolt's "Put-That-There" system [56]. Combined pointing and speech inputs offered a natural way to communicate, and later authors added gaze direction tracking to disambiguate other modalities [56]. Other early systems used speech input along with keyboard and mouse in an effort to support greater expressive power for complex visual manipulation. Speech technology advances in late 1980s allowed speech to become an alternative to keyboard leading to map and tourist information systems like CUBRICON [226] and Georal [296].

Bimodal systems that combine speech and pen-input or speech and lip-movements emerged in 1990s leading to work on integration and synchronization issues and the development of new architectures to support them. Speech and pen-input (2D or 3D gestures) involving hundreds of different interpretations beyond pointing have advanced rapidly leading to mature research (e.g. Quickset [86]) and commercial systems. Speech and lip movement systems exploit the detailed classification of human lip movements (visemes) and viseme-phoneme mappings that occur during articulatory speech. Lip movement offers speech recognition robustness in noisy environments and animated character systems with coordinated text-to-speech output and lip movement.

Examples of such systems include (*talking heads* or *speaking agents*) include the Rea system [79], KTH's August, Adapt and Pixie systems ([128, 127] and [126]). These systems use audiovisual speech synthesis and anthropomorphic figures to convey facial expressions and head or body movements. Systems with animated interactive characters have also been constructed such systems built at DFKI (see [38]). These systems mainly focus on multimedia presentation techniques and agent technologies. Information kiosks (*intelligent kiosks*) such as SmartKom ([321]) project use speech and haptics to provide interface for users in public places (e.g. museums).

Recently, systems combining 3 or more modalities such as person identification and verification systems which use both physiological (retina, fingerprints) and behavioral (voice, handwriting) modalities have been developed. Also there is an increased interest in *passive input modes* which refer to naturally occurring user behavior that is unobtrusively monitored by a computer (e.g., facial expressions). *Ambient intelligence* and blending of active and passive modes is a promising direction to this end.

6.3 Multimodal interaction

When the results of multiple modalities are combined, fusion techniques are needed for their integration. Early multimodal interfaces were based on a specific control structure for multimodal fusion. For example Bolt's demo, searches for a synchronized gestural act that designates the spoken referent. To support more broadly functional multimodal systems though, general processing architectures have been developed which handle a variety of multimodal integration patterns and support joint processing of modalities.

6.3.1 Multimodality levels

According to [231], we can differentiate four different uses of multimodal inputs and outputs depending on fusion type and use of modalities: *exclusive* (independent fusion, sequential modalities), *concurrent* (independent fusion, parallel modalities), *alternate* (combined fusion, sequential modalities) and *synergistic* (combined fusion, parallel modalities).

	use of modalities	
fusion type	Sequential	Parallel
Independent	**Exclusive**	**Concurrent**
Combined	**Alternate**	**Synergistic**

Table 2: Levels of multimodality

The *exclusive* use of modalities is the most straightforward, since independent modalities can be used at different times. This mode imposes the least requirements for a multimodal system. With *concurrent* fusion, modalities are used concurrently but their results are not combined in any way (they can for example be used for different tasks). Conversely, modalities are *alternative* when they are used at different times but their results are combined in some way. Finally, with synergistic use (the most sophisticated use of multimodality), modalities are combined at the same time. This puts heavy demands on the system and is seldom used.

6.3.2 Fusion techniques

Multimodal systems usually integrate signals at the feature level (*early fusion*) or at a higher semantic level (*late fusion*). In an early fusion architecture, the signal-level recognition process in one mode influences the course of recognition in the other and so, is considered more appropriate for closely temporally synchronized input modalities, such as speech and lip movements. Systems using the late fusion approach have been applied to processing multimodal speech and pen input or manual gesturing, for which the input modes are less coupled temporally and provide different but complementary information. Late semantic integration systems use individual recognizers that can be trained using unimodal data and can be scaled up easier in number of input modes or vocabulary size.

Alternatively, one can consider fusion at *lexical, syntactic* or *semantic* levels. Lexical fusion is used when hardware primitives are mapped to application events, syntactic fusion synchronizes different modalities and forms a complete representation of these and semantic fusion represents functional aspects of the interface by defining how interaction tasks are represented using different modalities.

6.3.3 Integration techniques

Multimodal systems based on late (semantic) fusion integrate common meaning representations derived from different modalities into a combined final interpretation. This requires: a common meaning representation framework for all modalities used and a well-defined operation for integrating the *partial meanings*.

Meaning representation uses data structures such as *frames* [217] and *feature structures* [171] or *typed feature structures* [77]. Frames represent objects and relations as consisting of nested sets of attribute/value pairs while feature structures goes further to use shared variables to indicate common substructures. Typed

feature structures are pervasive in natural language processing, and their primary operation is unification, which determines the consistency of two representational structures and, if they are consistent, combines them.

Various integration techniques have been derived so far: *frame-based integration* techniques use a strategy of recursively matching and merging attribute/value data structures (e.g., [289]) while *unification-based integration* techniques use *logic-based* methods for integrating the *partial meaning fragments*. Unification-based architectures have only recently been applied to multimodal system design [162, 161]. Some important unification-based integration techniques include feature-structure and symbolic unification. *Feature-structure unification* is considered well suited to multimodal integration, because unification can combine complementary or redundant input from both modes, but it rules out contradictory input. *Symbolic unification* which combined with statistical processing techniques results *hybrid symbolic/statistical* architectures which represent a new direction for multimodal system development and achieve very robust functioning, compared with either an early or late fusion approach alone.

6.3.4 Fission techniques

Fission is a process in which modalities are selected for outputs. For example, in multimodal speech systems outputs can be expressed by using synthesized speech, non-speech audio, text or graphics. Fission techniques have not gained as much attention as fusion techniques, and this is often thought of as a simple practical issue. Work has been done mainly in the context of multimedia systems, for example in the area of automated multimedia systems ([38]). The focus in these systems is often more on the rendering of the information for different medias than in the selection of the media for different elements.

6.4 Dialogue management

The dialogue manager controls the overall interaction between the system and the user by finding suitable system actions which corresponds to the user input, which can be seen as a mapping from a user action to a system action, or from one system state to an another system state. Many dialogue managers (especially those used in text based systems), tend to extend their functionality to natural language understanding and generation as well. The communication with the data source such as the database is often one of the tasks of the dialogue manager.

Although dialogue management is a fairly mature research area, and many sophisticated text-based systems have been constructed, they have not been proven to be very successful. Although speech is different from text, many of the principles found in these systems can be used in speech dialogue systems.

6.4.1 Dialogue initiative strategies

One of the key aspects in dialogue management is how the initiative is handled. The dialogue management strategy used may be system-initiative, user-initiative or mixed-initiative. We briefly review and compare each strategy in this section.

System-initiative dialogue strategy

With system-initiative dialogue, the computer asks questions from the user, and when the necessary information has been received, a solution is computed and a response is produced. It can be highly efficient since the paths which the dialogue flow can take are limited and predictable. It is most suitable for well-defined, sequential tasks where the system needs to know certain pieces of information in order to perform a task (e.g. a database queries for bus timetables or flights).

One key advantage is the predictable nature of the dialogue flow, which makes it possible to use context-sensitive recognition grammars, for every dialogue state, helping the recognizer to achieve more robust recognition results. Also, since the system asks questions, it can easier guide the user to reach his/her goal, making sure all necessary steps will be performed. This makes the user feel comfortable with the system and prevent disorientation (specially for the novice user). The main disadvantage is the clumsiness of interaction with experienced users, because only single pieces of information are exchanged in every dialogue turn, making the

dialogue advance slowly. The system may let experienced users pass certain dialogue turns by using more complicated expressions, but this may make the dialogue management complicated and recognition grammars more complex.

User-initiative dialogue strategy

User-initiative dialogue strategy assumes that the user knows what to do and how to interact with the system. The system waits for user inputs and reacts to these by performing corresponding operations. The main advantage is that experienced users are able to use the system freely and perform operations any way they like without the system getting in their way. This is also natural in open-ended tasks which have many independent subtasks. The main weakness is that they assume (and require) that users are familiar with the system and know how to speak. This imposes very open language models to the system and cognitive load to the user, which are both difficult to handle.

Mixed-initiative dialogue strategy

There is no single dialogue management strategy which is suitable for all situations. Different users and application domains have different needs, and different dialogue handling strategies may needed even inside a single application. With mixed-initiative strategy, the initiative can be taken either by the user or the system. The user has freedom to take the initiative, but when there are problems in the communication, or the task requires it, the system takes the initiative and guides the interaction. If properly constructed, a mixed- initiative system can help the user by employing system-initiative strategy while still preserving the freedom and efficiency of user-initiative strategy.

6.4.2 Dialogue control models

Various models have been devised for the overall structure of the dialog flow, like event and plan based, agent-based or even theorem-proving ones. In practice however models based on finite-state machines and frame-based are the ones most usually used.

Finite-state machines

Most of the current commercial speech applications use finite-state machines for dialogue control, because they are well known and straightforward to use. Finite-state machine consists of a set of nodes representing dialogue states and a set of arcs between the nodes, which move the dialogue from one state to another. The resulting network represents the whole dialogue structure, and paths through the network represent all the possible dialogues which the system is able to produce. If there are numerous states, and a lot of transitions between states, the complexity of the dialogue model increases rapidly, so they are mostly suitable for small-scale and system-initiative applications

Frame-based systems

Frame-based systems use collections of information (templates) as a basis for dialogue management and the purpose of the dialogue is to fill necessary information slots and then perform a query or similar operation on the basis of the frame. In contrast to the state-machine approach, they are more open, since there is no predefined dialogue flow (dialogue control is centralized and usually specified with a single algorithm), but instead the required information is fixed. Multiple slots can be filled by using a single utterance, and the order of filling the slots is usually free. It is a more natural choice for implementing mixed-initiative dialogue strategy, since the computer may take the initiative by simply asking for the required fields. VoiceXML applications (4.4.1) for example, use the frame-based approach for standard dialogue control and the event-based one for cases like error handling.

6.5 Architectures

Software architecture in multimodal systems was considered to be mostly a practical issue and it was often not modeled explicitly. This has already been noticed [213] and as such systems become more complicated, even more focus will be needed on system architectures. This is important, since systems can be more efficient and easier to build and maintain if proper architectural models are used.

Most systems are very complex in terms of architecture and software design, so they usually mix and exploit many software architectural styles and models like the pipe-and-filter, finite-state machine, event-based model, client-server, object-oriented and agent-based ones. For example spoken dialogue systems are usually structured either in a pipeline fashion or using the client-server model with a central component, which facilitates the interaction between other components, like the Galaxy-II architecture [287, 288]. Multimodal systems are based on even more sophisticated architectures like [183] or agent architectures (see 6.5.2).

In this paragraph we briefly examine the differences in requirements between GUI and multimodal architectures, then we turn our attention to the most common style of multimodal architectures and conclude with the development of multimodal frameworks which facilitate easier development of such complex applications.

6.5.1 GUIs vs Multimodal architectures

In Contrast with GUIs which assume that there is a single event stream that controls the underlying event loop, multimodal interfaces process continuous and simultaneous input from parallel incoming streams. Also GUIs assume that the basic interface actions, such as selection of an item, are atomic and unambiguous events, while multimodal systems process input modes using recognition-based technologies, which are designed to handle uncertainty and entail probabilistic methods of processing. Finally, multimodal interfaces that process two or more recognition-based input streams require time-stamping of input, and the development of temporal constraints on mode fusion operations.

6.5.2 Multimodal agent-based architectures

The most common infrastructure that has been adopted by the multimodal research community involves *multi-agent architectures*, such as the *Open Agent Architecture* [203] and *Adaptive Agent Architecture* [183]. Multi-agent architectures provide essential infrastructure for coordinating the many complex modules needed to implement multimodal system processing, and they permit doing so in a distributed manner. In a multi-agent architecture, the many components needed to support the multimodal system (e.g., speech recognition, gesture recognition, natural language processing, multimodal integration) may be written in different programming languages, on different machines, and with different operating systems. Agent communication languages are being developed that can handle asynchronous delivery, triggered responses, multi-casting and other concepts from distributed systems.

Using a multi-agent architecture, for example, speech and gestures can arrive in parallel or asynchronously via individual modality agents, with the results recognized and passed to a *facilitator*. These results, typically an nbest list of conjectured lexical items and related time-stamp information, then are routed to appropriate agents for further language processing. Next, sets of meaning fragments derived from the speech, etc. arrive at the multimodal integrator which decides whether and how long to wait for recognition results from other modalities, based on the system's temporal thresholds. It fuses the meaning fragments into a semantically-and temporally-compatible whole interpretation before passing the results back to the facilitator. At this point, the system' s final multimodal interpretation is confirmed by the interface, delivered as multimedia feedback to the user, and executed by any relevant applications.

6.5.3 Multimodal frameworks

Despite the availability of high-accuracy speech recognizers and the maturing of multimodal devices such as gaze trackers, touch screens, and gesture trackers, very little applications take advantage of these technologies. One reason for this may be that the cost in time of implementing a multimodal interface is prohibitive. One desiring to equip an application with such an interface must usually start from scratch, implementing access to external sensors, developing ambiguity resolution algorithms, etc. However, when properly implemented, a large

part of the code in a multimodal system can be reused. This aspect has been identified and many multimodal application frameworks have recently appeared such as VTT's *Jaspis* and *Jaspis2* frameworks [312, 311], Rutgers CAIP Center framework [109], the embassi system [103] and more.

6.6 Standards

6.6.1 W3C standards

The number of different kinds of devices that can access the Web has grown from a small number with essentially the same core capabilities to many hundreds with a wide variety of different capabilities like mobile phones, smart phones, personal digital assistants, kiosks, automotive interfaces, etc.

Device Independence

The range of capabilities for input and output and the range of markup languages and networks supported greatly complicate the task of authoring web sites and applications that can be accessed by users whatever device they choose to use. The W3C *Device Independence Working Group* encompasses the techniques required to make such support an affordable reality. In particular the activity focuses on methods by which the characteristics of the device are made available for use in the processing associated with device independence methods to assist authors in creating sites and applications that can support device independence in ways that allow it to be widely employed. The group has overtaken work by *Composite Capability/Preference Profiles Working Group* (CC/PP see [22]), and through coordination with *Web Accessibility Initiative* [29] and *MultiModal Interaction Working Group* [25] continues it's work on avoiding the fragmentation of the Web into spaces that are accessible only from subsets of devices.

Multimodal Interaction Activity

Mobile profiles have emerged using a number of W3C specifications like XHTML, making mobile access more close to reality. Recently, a tremendous growth of interest in using speech as a means to interact with Web-based services over the telephone (*Voice Browser Activity*), but spoken interfaces (based upon VoiceXML), only prompt users with pre-recorded or synthetic speech and understand simple words or phrases. There is now an emerging interest in richer forms of interaction, combining speech with other modalities. Multimodal interaction will enable the user to speak, write and type, as well as hear and see using a more natural user interface than today's single mode browsers.

The *Multimodal Interaction Activity* is extending the Web user interface to allow multiple modes of interaction(aural, visual and tactile), offering users the means to provide input using their voice or their hands via a key pad, keyboard, mouse, or stylus. For output, users will be able to listen to spoken prompts and audio, and to view information on graphical displays. By allowing multiple modes of interaction (GUI, speech, vision, pen, gestures, haptic interfaces, etc), to any device it facilitates the dream of *accessibility to all.*

The Working Group was launched in 2002 following a joint workshop between the W3C and the WAP Forum with contributions from SALT [17] and XHTML+Voice (X+V) [14]. It's major contributions include: *Multimodal Interaction Use Cases*, *Multimodal Interaction Use Requirements*, the *W3C Multimodal Interaction Framework* [26]. Work has also been done on dynamic adaptation to device configurations, user preferences and environmental conditions (*System and Environment Framework*) [27], on integration of composite multimodal input and modality component interfaces such as interfaces for ink and keystrokes which will enable the use of grammars for constrained input, and the context sensitive binding of gestures to semantics (speech and DTMF modalities are developed by the *Voice Browser Working Group* [28]).

Group's work has also stimulated the creation of mark-up languages such as EMMA, and InkML. EMMA (*Extensible MultiModal Annotation Markup Language*) [23], formerly known as *Natural Language Semantics Markup Language*, is a markup language intended to represent semantic interpretations of user input (speech, keystrokes, pen input etc.), together with annotations such as confidence scores, timestamps, input medium etc. The interpretation of the user's input is expected to be generated by signal interpretation processes, such as speech and ink recognition, semantic interpreters, and other types of processors for use by components that act on the user's inputs such as i interaction managers. InkML [24], defines an XML data exchange format for

ink entered with an electronic pen or stylus as part of a multimodal system, which will enable the capture and server-side processing of handwriting, gestures, drawings and other specific notations.

6.6.2 Salt and X+V

Until W3C standards emerge and mature, other related efforts have been shown, namely SALT and XHTML + Voice. *Speech Application Language Tags* (SALT) [17], is a lightweight set of extensions to existing markup languages, allowing developers to embed speech enhancements in existing HTML, XHTML and XML pages, enabling multimodal and telephony-enabled access to information, applications, and Web services from PCs, telephones, tablet PCs, and PDAs. *XHTML+Voice* [14], by IBM, Motorola and Opera Software, is yet another effort exploiting the combined use of XHTML and parts of VoiceXML through *XML events* to support for visual and speech interaction. In contrast with SALT, X+V provides a standard visual markup language (XHTML) and an event model, has reacher voice interaction and makes development easier by allowing separation of visual and voice programming. Development tools from IBM are already available and so is a multimodal browser from Opera for Sharp's Zaurus PDA.

7 Eye Tracking Interfaces

Research activity in eye tracking and visual attention has increased in the last few years due to improvements in performance and reductions in the costs of eye tracking devices as well as better understanding of the human visual system. Three types of movements need be modeled to gain insight into the overt localization of gaze. These are the primary requirements of eye movement analysis: the identification of fixations, saccades, and smooth pursuits. Fixations naturally correspond to the desire to maintain one's gaze on an object of interest. Eye pursuits follow objects in smooth motion. Saccades are the expression of the desire to voluntarily change the focus of attention.

7.1 Eye Tracking Technology

A number of eye gaze detection methods have been developed over the years. There are four broad categories of eye movement measurement methods, involving the use of: electro-oculography (EOG), scleral contact lens/search coil, photo-oculography (POG) or video-oculography (VOG), and video-based combined pupil and corneal reflection. Electro-oculography, or EOG, relies on DC recordings of the electric potential differences of the skin surrounding the ocular cavity. Invasive methods that required tampering directly with the eyes were mostly used before the 70s. The search coil method [270] offers high accuracy and large dynamic range but requires an insertion into the eye. Non-invasive methods such as the DPI (Dual Purkinje Image) eye tracker [57] require the head to be restricted and are relatively expensive. More recently systems have appeared that use video images and/or infrared cameras.

Several methods of improving the accuracy of estimating gaze direction and inferring intent from eye movement have been proposed. The Eye-R system [286] is designed to be battery operated and is mounted on any pair of glasses. It measures eye motion using infrared technology by monitoring light fluctuations from infrared light and utilizes this as an implicit input channel to a sensor system and computer. As a person walks around, information is exchanged between the Eye-R module and the exhibit that the user fixates on. This information is transferred to a server using a network or through the glasses. All exhibits are fitted with an infra-red sensor in this networked environment.

Some commercial manufacturers of eye trackers now have a head-based system that has an attached miniature camera that takes an instant picture of the scene once a fixation is detected. Mulligan [221] uses a low cost approach to track eye movement using compressed video images of the fundus on the back surface of the eyeball. It is capable of high performance as off-line data analysis is acceptable. More accurate results can be obtained when the imagery is analyzed off-line using more complex algorithms implemented in software. A technical challenge for these types of trackers is the real time digitization and storage of the video stream from the cameras. New video compression technology allows streams of video images to be acquired and stored on normal computer system disks; however lossy compression can lead to loss of important information.

Bhaskar et al [54] propose a method that uses eye blink detection to locate an eye which is then tracked using an eye tracker. Blinking is necessary for the tracker to work well and the user has to be aware of this.

ASL [7], Smarteye [8], IBM's Blue Eyes [9], Arrington's Viewpoint [10], SR's Eyelink [11], EyeGaze [12] and CRS [13] eye trackers are examples of recent commercial eye trackers. A typical commercial eye tracker tracks the pupil and the first purkinje image (corneal reflex) and the difference gives a measure of eye rotation.

Most eye trackers require calibration because of individual differences of eyeball size and the difficulty in measuring the position of the fovea. The FreeGaze System [308] attempts to limit errors arising from calibration and gaze detection by using only two points for individual personal calibration. The position of the observed pupil image is used directly to compute the gaze direction but this may not be in the right place due to refraction in the surface of the cornea. The eyeball model corrects the pupil position for obtaining a more accurate gaze direction.

7.2 Human Gaze Behaviour

Experiments have been conducted to explore human gaze behaviour for different purposes. Privitera et al [262] use 10 image processing algorithms to compare human identified regions of interest with regions of interest determined by an eye tracker and defined by a fixation algorithm. The comparative approach uses a similarity

measurement to compare 2 aROIs (algorithmically-detected Region of Interests), 2 hROIs (human-identified Region of Interests) and an aROI plus hROI. The prediction accuracy was compared in order to identify the best matching algorithms. Different algorithms fared better under differing conditions. They concluded that aROIs cannot always be expected to be similar to hROIs in the same image because 2 hROIs produce different results in separate runs. This means that algorithms are unable in general to predict the sequential ordering of fixation points.

Jaimes, Pelz et al [155] compare eye movement across categories and link category-specific eye tracking results to automatic image classification techniques. They hypothesize that the eye movements of human observers differ for images of different semantic categories, and that this information can be effectively used in automatic content-based classifiers. The eye tracking results suggest that similar viewing patterns occur when different subjects view different images in the same semantic category. Five different categories are considered: handshakes, crowds, landscapes, main object in uncluttered background and miscellaneous images. More consistent viewing patterns were found within the handshake and main object categories. Although, it was unclear how it can be used to influence automatic classification techniques, they suggested that it is possible to apply the Privitera's fixation clustering approach [262] to cluster gaze points. The study showed that similar viewing patterns can be category-specific hence this factor needs to be considered in future algorithms.

Pomplun and Ritter [256] present a three-level model, which is able to explain about 98% of empirical data collected in six different experiments of comparative visual search. Pairs of almost identical items are compared requiring subjects to switch between images several times before detecting a possible mismatch. The model consists of the global scan path strategy, shifts of attention between two visual hemifields, and eye movement patterns. Simulated gaze trajectories obtained from this model are compared with experimental data. Results suggest that the model data of most variables presents a remarkably good correspondence to the empirical data.

Identification and analysis of fixations and saccades in eye tracking protocol is important in understanding visual behaviour. Salvucci [277] classifies algorithms with respect to five spatial and temporal characteristics. The spatial criteria divide algorithms in terms of their use of velocity, dispersion of fixation points, and areas of interest information. The temporal criteria divide algorithms in terms of their use of duration information and their local adaptivity. It was concluded that velocity-based and dispersion-based algorithms fared well and provided similar performance.

	Fixation Identification Algorithms				
Criteria	Velocity Threshold	Hidden Markov Model	Dispersion Threshold	Minimum Spanning Tree	Area of Interest
Velocity-based	X	X			
Dispersion-based			X	X	
Area-based					X
Duration-sensitive			X		X
Locally-adaptive		X	X	X	

Table 3: Identification and analysis of fixations and saccades in eye tracking

The five fixation identification algorithms are also described and compared in terms of their accuracy, speed, robustness, ease of implementation, and parameters. The results show that hidden markov models and dispersion based algorithms fare better in terms of their accuracy and robustness. The Minimum Spanning Tree uses a minimized connected set of points and provides robust identification of fixation points, but runs slower due to the two step approach of construction and search of the minimum spanning trees. The velocity threshold has the simplest algorithm and is thus fast but not robust. Areas of Interest are found to perform poorly on all fronts. These findings are implemented in the EyeTracer system [276], an interactive environment for manipulating, viewing, and analyzing eye-movement protocols. EyeTracer facilitates both *exploratory analysis* for initial understanding of behaviours and model prototyping and *confirmatory analysis* for model comparison

and refinement.

NASA's Lee Stone [190] focuses on the development and testing of human eye-movement control with particular emphasis on search saccades and the response to motion (smooth pursuit). The specific goal is to incorporate recently acquired empirical knowledge of how eye movements contribute to information gathering, and of the relationship between the eye movement behaviour and the associated percept, into computational tools for the design of more effective visual displays and interfaces that are matched to human abilities and limitations. Much of the focus is on proposing a new control strategy for pursuit eye movement modified from an existing model. Stone concludes that current models of pursuit should be modified to include visual input that estimates object motion and not merely retinal image motion as in current models.

Duchowski [97] presents a 3D eye movement analysis algorithm for binocular eye tracking within Virtual Reality. The signal analysis techniques are given three categories: position-variance, velocity-based and ROI-based, again using two of Salvucci's criteria [277]. The algorithm uses velocity and acceleration filters for eye movement analysis in three-space. This is easily adapted to a 2D environment by holding head position and visual angle constant. Gazepoints in the virtual environment are calculated by the 2D to 3D mapping of gaze vectors. The computed gaze direction vector is use for calculating gaze intersection points. The algorithm is evaluated using a virtual environment for aircraft visual inspection training. It was concluded that cognitive feedback, in the form of visualized scan-paths, does not appear to be any more effective than performance feedback (search timing). Also, the number of fixations decreases following training.

7.3 Visual Attention Modelling

Visual attention is a capability that allows us to focus on objects that are likely to be of interest whilst ignoring background material that is of no relevance. It is a reasonable assumption that gaze behaviour will be influenced to a large extent by the saliency of various regions in images as our eyes will be drawn towards those parts that stand out or are anomalous in some way. Gaze will also be affected by the interests and motivation of the viewer which may be more difficult to anticipate. Studies of visual attention models will be carried out in Work Package 5 sub task 4 (Saliency detection, visual features and their organization).

7.4 Eye Tracking Interfaces

The best interfaces are the most natural ones. They need to be unobtrusive and provide relevant information quickly and in ways that do not interfere with the task itself. Eye tracking offers new ways of measuring human behaviours and is being used in both active and passive modes in several applications.

There have been many applications of eye tracking technology including a replacement for a mouse [131], a tool to vary the screen scrolling speed [233] and a way of assisting disabled users [90]. Schnell and Wu [282] apply eye tracking as an alternative method for the activation of controls and functions in aircraft. In another domain marketing researchers frequently determine what features of product advertisements attract buyers' attention and alter designs accordingly [43].

The Dasher system [324] captures text using a method that relies purely on gaze direction. The user composes text by looking at characters as they stream across the screen from right to left. Dasher presents likely characters in sizes according to the probability of their occurrence in that position. The user is often able to select rapidly whole words or phases as their size increases on the screen. In comparison with on-screen keyboards, it is not confounded by the problem of random, but natural eye movements, the so-called "Midas-Touch".

Nikolov et al [232] propose a system for construction of gaze-contingent multi-modality displays of multi-layered geographical maps. Gaze contingent multi-resolutional displays (GCMRDs) position high-resolution information around the user's gaze position, matching the user's interest. In this system, different map information is channeled to the central and peripheral vision giving real performance advantage.

Xin Fan et al [330] propose an image viewing technique based on an adaptive attention shifting model, which addresses the problem of browsing large images on the small screens of mobile phones. Maximum use of screen area is achieved by cropping images to those areas on which viewers are likely to fixate.

Following experiments on eye gaze behaviour Farid [107] concluded that systems performed well because of the minimal latency and obtrusiveness. He also noted that dwell time was an unreliable indicator of user

interest. A zooming technique was adopted that provided a magnified region of interest and multiple video streams.

7.5 Shortcomings

The recent advances in eye tracking technology have played a large part in encouraging more research into image analysis that take account of human behaviour. More accurate and precise eye tracking experiments are leading to a better understanding of the human visual system (HVS) from which new models are derived and new applications are stimulated.

Two main shortcomings have been identified with eye tracking system and attempts have been made with varying success to combat these drawbacks. Firstly, eye tracking hardware systems must limit processing to attain real-time performance whilst maintaining maximum accuracy of the eye movement measurement [221]. Blinking has been suggested for rapid eye localisation [54] and solving the Midas-touch problem [100], but this puts a great burden upon the user. The use of a second device, such as a mouse, is distracting and negates the need for interfaces controlled by the eyes in the first place. Fixation dwell times are simple to extract and perhaps might be thought to yield more stable data, however, these have been shown to be unreliable indicators of regions of interest.

Secondly, even though eye fixations provide some of the best measures of visual interest, they do not necessarily provide a measure of cognitive interest. Although eye tracking offers an objective view of overt human visual and attentional processes, it does not provide a measure of covert attention. Human visual attention is frequently not always concentrated at the centre of vision, but often several degrees to one side. This effect must be recognised in any application.

Many of the recent eye tracking investigations have concentrated upon replacing and extending existing computer interface mechanisms rather than creating a new form of interaction. Gaze normally reflects our intentions and desires and so data obtained by eye tracking could be used to provide a more intimate form of human computer interaction. However, the imprecise and unpredictable nature of saccades and fixation points has prevented eyetracking from yielding large benefits over conventional human interfaces. Fixations, saccades and smooth pursuit vary considerable from person to person which means that statistical approaches to interpretation (such as clustering, summation and differentiation) are inadequate for identifying salience in an image. More robust methods of interpreting eye tracking data are needed.

7.6 Future Work and Relationship to WP10

Eye tracking technology is now a serious contender for new interfaces for accessing digital visual data. Patterns of fixations and saccades certainly reflect image content and this information could be used not only to categorise the material but also to infer the viewer's intentions.

We plan to conduct experiments that relate eye behaviour to models of visual attention to determine the feasibility of predicting fixation points while browsing an image database. Gaze information will be gathered during task oriented search that will explore the variability of user behaviour and again provide information for use in predicting behaviour during similar searches.

This work will lead naturally into a CBIR demonstrator that will show how a linked set of similar regions across images in a data base may be traversed during a search driven by an eye tracker. Work carried out in Work Package 5 sub task 4 (Saliency detection, visual features and their organization) will investigate automatic techniques for setting up a network of visual similarity associations between images, although these may initially be constructed manually for a small dataset in this task.

8 Audio-Visual Speech Recognition and Interfaces

Commercial *Automatic Speech Recognition* (ASR) systems are uni-modal, i.e., only use features extracted from the audio signal to perform recognition. Although audio-only speech recognition is a mature technology with a long record of significant research and development achievements [264], current uni-modal ASR systems can work reliably only under rather constrained conditions, where restrictive assumptions regarding the size of the vocabulary, the amount of noise etc can be made. These shortcomings have seriously undermined the role of ASR as a pervasive *Human-Computer Interaction* (HCI) technology and have limited the applicability of speech recognition systems to well-defined applications like dictation and low-to-medium vocabulary transaction processing systems.

On the other hand, speech recognition by humans is fundamentally multi-modal. Although audio is the most important source of information for speech recognition, people also use visual cues as a complimentary aid in order to successfully perceive speech. The key role of the visual modality is apparent in situations where the audio signal is either unavailable or severely degraded, as is the case with hearing-impaired listeners or very noisy environments, where seeing the speaker's face is indispensable in recognizing what has been spoken. Human perception weighs the visual information more when two articulated sounds are not easily discernible acoustically, but can be discriminated visually due to a different place of articulation, as is the case with the phonemes /n/ and /m/, which sound very similar but look quite different [260]. This phenomenon is lucidly manifested in a well known psychological illusion, the so-called *McGurk effect* [212, 204]. In their experiments, McGurk and MacDonald found out that when somebody experiences contradictory audio and visual speech cues, he/she tends to perceive whatever is most consistent with both sensory information. The McGurk effect shows that human speech understanding is multi-modal, resulting from sensory integration of audio and visual stimuli.

These findings provide strong motivation for the Speech Recognition community to do research in exploiting visual information for speech recognition, thus enhancing ASR systems with speech-reading capabilities [301]. Research in this relatively new area has shown that multimodal ASR systems can perform better than their audio-only or visual-only counterparts. The first such results where reported back in the early 80's by Petajan [251]. The performance gain becomes more substantial in scenarios where the quality of the audio signal is degraded, as is the case with particularly noisy environments such as a vehicle's cabin [252].

However, the design of robust audio-visual ASR systems, which perform better than their audio-only analogues in all scenarios, poses new research challenges. Two new major issues arise in the design of audio-visual ASR systems, namely:

- *Selection and robust extraction of visual speech features.* From the extremely high data rate of the raw video stream, one has to choose a small number of salient features which have good discriminatory power for speech recognition and can be extracted automatically, robustly and with low computational cost.
- *Optimal fusion of the audio and visual features.* Inference should be based on the heterogenous pool of audio and visual features in a way that ensures that the combined audiovisual system outperforms its audio-only counterpart in practically all scenarios. This is definitely non-trivial, given that the relative quality of the audio and visual features can vary dramatically during a typical session.

A block diagram of an audiovisual ASR system depicting its main components is shown in Fig. 3.

In the sequel, we will attempt to review the main trends in the literature for addressing the two aforementioned major research challenges in *Audiovisual ASR* (AV-ASR), namely the design of the visual front-end and the fusion of audio-visual features. These two issues are by no means trivial to solve and are the subject of current intensive research. Other reviews of the subject are [261, 301]. This is a short version of our report that appears in the Work Package 6 deliverable.

8.1 The Visual Front End of Audio-Visual Automatic Speech Recognition Systems

As emphasized in Section 8, the design of the visual front end is one of the main challenges in building an audiovisual ASR system. One can generally identify the following steps in the processing of the visual modality input [134, 261]:

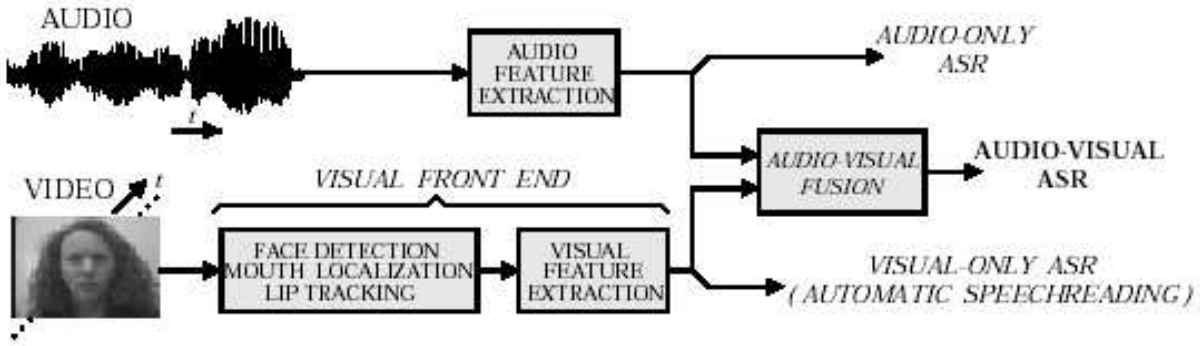

Figure 3: Block diagram of a typical audiovisual ASR system. Figure from [261].

- *Active speaker's face detection and tracking.* The speaker's face contains significant information for visual speech-reading and thus the system must reliably locate and track it.
- *Facial model fitting.* In the case an exemplar is used to model the lips' borders (e.g. an active contour model [168]) or even the whole face (e.g. an active appearance model [88]), the exemplar is roughly initialized near the identified *Region Of Interest* (ROI) and evolves towards its final best-fit configuration.
- *Visual features extraction.* After the ROI has been identified, a number of visual features are extracted from it. These features can be appearance-based, shape-based, or a combination of them.

We proceed by examining the main methodologies to address these problems.

8.1.1 Active speaker's face detection and tracking

In order for the visual modality to be useful for the task of speech recognition, an AV-ASR system must be able to reliably detect and track the speaker's face, which contains important visual cues for speech recognition. This task usually involves detecting human faces in the first video frame which are used in the sequel to initialize a face tracker. The detector is typically triggered again periodically to accommodate for a possible tracking failure and detect any new face entering the scene in the meantime. In the case of multiple persons being present in the scene, an additional requirement for the visual front-end is to determine who of them is the active speaker. Therefore the main tasks that the visual front-end of an AV-ASR system needs to do in order to detect and track the speaker's face are *face detection*, *face tracking* and *active speaker detection*.

Human **face detection** in still images is one of the main problems in Computer Vision, arising in important applications, like face recognition, area surveillance and Human–Computer Interaction. We will review here the main methods for face detection based on the statistical learning paradigm. For another recent survey consult [331].

Schneiderman and Kanade [280] apply statistical likelihood tests, using feature output histograms. Sung and Poggio [302] model faces and non-faces as mixtures of anisotropic Gaussians in a high-dimensional linear space. Rowley and Kanade [274] use neural network-based filters to create an excellently performing (but slow) face detection system. In Papageorgiou et al. [243] a general object detection scheme which uses a wavelet representation and statistical learning techniques is proposed. Osuna et al. [237] apply support vector machines as classifiers for face detection, and Romdhani et al. [271] improve on that work by devising a faster learning algorithm. A breakthrough in the speed of face detection algorithms, with a system able to process 15 images per second (an order of magnitude improvement over previous implementations of face detection algorithms) is reported by Viola and Jones [319]. They achieve this introducing the integral image representation, which allows for rapid computation of their rectangular features, in a system that uses the extremely efficient boosting method as classifier. Another system based on the boosting paradigm, which can also detect profile faces, is reported by Li and Zhang in [193].

After a face has been detected in a frame, a **face tracking** module is needed to track it for the subsequent frames until the face detector is triggered again. This processing step is only needed when the face detector

is not activated every single frame. The tracker might be designed to track either the speaker's lips only or his/her whole face, depending on the modeling approach. Since this is tightly connected with the model used to describe the speaker's face, we defer discussing the issue until 8.1.2.

The **detection of the active speaker's face** (in contrast to non-speakers' faces) in the case that many people are present in the visual scene is the final step to be done in order to successfully locate and track the speaker's face. This is especially important for the deployment of AV-ASR systems in realistic environments, like meeting rooms with many participants, where a number of attendants speak one after each other and the system needs to discern who is the active speaker at each particular moment. The same requirement is also posed by other applications, such as tele-conference systems, where the camera needs to zoom on the active speaker [91].

For that purpose a number of techniques have been devised. While early attempts were based on audio-only sound source localization techniques, most of the recent approaches to the problem utilize both audio and visual cues to successfully identify the speaker among the different persons present in the scene. The resulting fused system can be more robust to both vision and audio background clutter than corresponding single-modal systems. Two such techniques, presented in [47] and [81]. Other relevant references are [317], [339] and [137].

8.1.2 Facial model fitting

After the speaker's face has been located, speech related information must be extracted from it. There are generally two approaches for achieving this goal. The first, model-free approach is to find a rectangular ROI around the mouth area and subsequently use a transformation of the raw pixel values in this ROI as a feature set. This approach will be discussed further in 8.1.3. The second, model-based approach is to try and match a facial shape or appearance model to the observed face. The parameters of this model can be used in the sequel for creating the feature set. In this subsection we will describe some representative approaches used to model the face or parts of it.

Active contours (also called snakes) [168, 151] have been used to model and track the speaker's lips [63, 169, 83]. Accurate temporal tracking is achieved by means of a Kalman filter or a particle filter, after learning the lip motion dynamics [150, 169]. It would be interesting to test in this problem the performance of non-parametric, geometric active contour models, such as geodesic active contours with shape priors [78, 192].

A related method to model parts of the face is through **deformable templates** [334, 130], utilized in the AV-ASR context in [134, 80]. Using deformable templates, the shape of the lips is modeled by a small number of curves, capturing the shape of the lips with very few parameters. The template is allowed to deform by minimizing an associated cost functional via gradient descent.

Another powerful approach to human face modeling is through **Active Shape Models** (ASMs) and **Active Appearance Models** (AAMs). In ASMs, which were first proposed in [89], an object's shape is modeled by a set of landmark points. The shape's main modes of variation are learned by means of a PCA analysis, using a training set of images where the landmarks are manually annotated. A local appearance profile in the neighborhood of each landmark is also learned during the training phase. ASMs were first applied to lipreading by [196]. AAMs [88] and the closely related methods of Morphable Models [163] and Active Blobs [284] are an extension of ASMs, in the sense that the main modes of variation of both the object's shape and appearance (after warped to the mean shape) are learned from the training data by applying PCA twice. A third PCA is sometimes applied to capture the correlations between shape and appearance parameters. The model fits a novel image by minimizing the appearance reconstruction error. By taking the appearance information into account, AAMs are more robust to the initial condition than ASMs. AAMs were first used for AV-ASR in [206], where it was shown that they perform better than ASMs in this task.

8.1.3 Visual Features

In order the visual information in a video stream depicting the speaker to be useful for speech recognition, a compact set of about 10-100 informative features must be extracted from each frame and be used later for statistical classification. These features should be as robust as possible when different people talk and their poses or lighting conditions vary. Unlike audio-only speech recognition, where the properties of various sets of features are well understood, research on the relative merits of alternative visual features for visual speech

recognition is far less mature. One can classify the various visual features proposed in the literature into three broad categories [261], namely:

- *appearance features*, which directly use (a transform of) the pixel values in the mouth ROI.
- *shape features*, where the parameters of a shape model are used to derive the features.
- *combined shape and appearance features*, where information from both the shape and the appearance of the ROI are used to form the features.

In the sequel we further discuss these approaches. One can refer to [261] for more details.

Appearance Features

This approach to feature extraction doesn't need a shape model to be fit on the speaker's face, as described in 8.1.2. It only requires that a ROI around the speaker's mouth has been identified at each video frame. The preliminary feature vector consists then of the concatenated greyscale or color pixel values, having length $d = N$ or $d = 3N$, respectively, where N is the number of pixels in the ROI. Since the length d of the preliminary feature vector is usually still large, a dimensionality reduction technique is usually applied, before these features are used for speech recognition.

One usual unsupervised learning approach to dimensionality reduction is through **Principal Component Analysis** (PCA) [200]. This method uses a training set of ROI images to learn an affine space of reduced dimensionality $k < d$ capturing the main modes of variation (eigenimages) in the class of ROI images. PCA-based dimensionality reduction has been used extensively in the context of speechreading (see the references in [261]). Another popular approach for coding visual features for visual speech recognition is to use image compression techniques, based on standard **image transforms**, such as the *Discrete Cosine Transform* (DCT) and the *Discrete Wavelet Transform* (DWT) [259].

A different, supervised learning approach for dimensionality reduction used often in practice is **Linear Discriminant Analysis** (LDA). LDA finds a projection matrix such that the between-class variance of the projected data is maximized relative to their within-class variance and can be shown to be the optimal decision rule in the case that the classes are Gaussian and have a common covariance matrix [98]. For more details and enhancements to LDA, see the references in [261].

Shape Features

This approach to feature extraction assumes that the shape of the ROI contains enough information for visual speech recognition. Shape based features are extracted utilizing shape models, which were described in 8.1.2. The features of interest are usually extracted from the shape of the speaker's lips area, although in some cases larger parts of the face are used [207]. The feature vector can either describe some geometric properties of the lip's shape or, alternatively, just consist of the parameters of the specific shape model used for shape fitting.

After the lip contours have been identified in the current frame with one of the methods discussed in Sec. 8.1, a vector of **lip geometric features** can be extracted from them to subsequently be used for visual speech recognition. Examples of such features, which have been used extensively by various AV-ASR systems, include the height, width, perimeter, as well as the area contained within the contour. Another approach to describe the shape of the lips succinctly is through lip shape moments or Fourier descriptors [96]. See [261] for more details.

If a parametric shape model has been fitted to the speaker's face, as described in 8.1.2, it is natural to use the **shape model parameters** as shape features. The parameters of Active Contours tracking the speaker's lips have been used for speech recognition in many speechreading systems, including the ones presented in [63, 83]. The variables controlling the shape of deformable templates fitted on the lips have been utilized for the same task in [134, 80]. Finally, the parameters of ASMs have been used for AV-ASR in [196, 206], among others.

Combined Shape and Appearance Features

Since appearance or shape only features are useful for visual speechreading, it seems plausible that features encoding both shape and appearance information can be most efficient in capturing visual information for

speech recognition. Some researchers have therefore tried to integrate joint shape and appearance features in their AV-ASR systems. In most early attempts to achieve this goal, features from each category are just concatenated. In [196], for example, shape parameters from a fitted ASM model were combined with the intensity profiles around each landmark of the ASM (see 8.1.2) in order to enhance the ASM with appearance information. The advent of statistical tools such as the AAMs, described in 8.1.2, which can model both the shape and the appearance of the face in a unified framework, has resulted in a more principled way to describe visual speech information with combined shape and appearance features. The first AV-ASR system utilizing AAMs in its visual front-end is reported in [206].

Visual Feature Comparison

An efficiency comparison between different sets of visual features is complicated, because the various researchers usually test their methods on different AV-ASR corpora and on different tasks, ranging from connected digit recognition to *Large Vocabulary Continuous Speech Recognition* (LVCSR).

Having said that, a couple of comparisons among different visual features are worth mentioning. First of all, a number of studies have shown appearance information, in the form of either appearance features or joint shape and appearance features, is indispensable in visual speech recognition. For example, AAMs outperform ASMs in the work reported in [206] and simple DCT appearance features give better results than lip contour geometric features in [259]. The experiment on speaker-independent LVCSR task documented in [207] shows that simple, image transform, appearance-based features, which require no particular training, perform better than AAMs, whose efficiency critically depends on careful training. For more references, one can consult [261].

8.2 Audio Visual Integration for Speech Recognition

To successfully address the problem of audiovisual speech recognition, it certainly does not suffice to complement the set of robust audio features with an informative set of visual features from the video stream, following the methods described in the previous section. The main task that needs to be addressed next is the fusion of the heterogenous pool of audio and visual features in a way that ensures that the combined audiovisual system outperforms its audio-only counterpart in all practical scenarios [33]. This task is complicated due to a couple of issues, the main of them being:

- Audio and visual speech asynchrony. Although the audio and visual observation sequences are certainly correlated over time, they exhibit state asynchrony, with visual activity preceding auditory activity by as much as 120 ms [63], close to the average duration of a phoneme. As we will see, this asynchrony renders modeling audiovisual speech with conventional HMMs [264] problematic.
- The relative speech discriminative power of the audio and visual streams can vary dramatically during a typical session in unconstrained environments, making their optimal fusion a challenging task.

Therefore successful audio and visual feature integration requires utilization of advanced techniques and models for cross-modal information fusion. This research area is currently very active and many different paradigms have been proposed for addressing the general problem. In the sequel we will confine ourselves to reviewing the main research trends for feature fusion in the context of audiovisual feature integration.

One can generally classify the various approaches to audio and visual feature integration into three main categories [134], depending on the stage that the audio and visual streams are fused, namely early, intermediate and late integration techniques. We will discuss next about the properties of these classes of methods.

8.2.1 Early Integration Techniques for Audio-Visual ASR

The simplest approach to audio-visual feature integration is through early integration methods. This class of techniques utilize a single classifier, avoiding the explicit modeling of the two different speech modalities.

In early integration approaches to audiovisual integration one simply concatenates the audio and visual feature vectors to obtain a single combined audiovisual vector [33]. In order to reduce the length of the resulting feature vector, dimensionality reduction techniques like LDA are usually applied before the feature vector finally feeds the recognition engine [260]. The classifier utilized by most early integration audiovisual ASR systems is a

conventional HMM, which is trained using the mixed audiovisual feature vector. Additional details on AV-ASR techniques based on audiovisual feature fusion can be found in [261].

8.2.2 Intermediate Integration Techniques for Audio-Visual ASR

Since early integration techniques avoid the explicit modeling of the multimodal nature of speech, they fail to model both the fluctuations in the relative reliability and the asynchrony problems between the two distinct audio and visual streams. In order to address these shortcomings, the multi-modality of audiovisual speech needs to be modeled more faithfully than conventional HMMs allow. A number of HMM extensions, belonging to the class of *Dynamic Bayesian Networks* (DBNs) [93], have been proposed in the literature in an attempt to address this goal. These models have two aspects in common, namely:

- They attempt to explicitly capture the reliability of each modality by letting the class conditional observation likelihood to be the product of the observation likelihoods of the single-stream components, raised to appropriate stream exponents that vary depending on the confidence of each stream.
- They allow modeling the state asynchrony of the audio and visual streams while preserving their natural correlation over time.

Some representative such models are the *multistream HMM* [333], the *product HMM*, the *factorial HMM* [119] and the *coupled HMM* [62]. A comparative study of the relative performance of these DBN-based architectures for the isolated word recognition task has shown that the C-HMM outperforms the other models in almost all cases [227].

8.2.3 Late Integration Techniques for Audio-Visual ASR

Late integration models utilize two independent HMMs, one for the audio and one for the visual features stream, which can be trained separately. The final classification decision is reached by combining the partial outputs of the uni-modal classifiers. The correlations between the visual and acoustic channels are not captured by these models.

In more detail, for small-vocabulary, isolated word speech recognition, late integration can be easily implemented by combining the audio- and visual-only log-likelihood scores for each word model in the vocabulary, given the acoustic and visual observations [33]. However, this approach is intractable in the case of connected word recognition or LVCSR, where the number of alternative paths explodes. A good heuristic alternative in that case is through lattice rescoring [333]. The n most promising hypotheses are extracted from the audio-only recognizer and they are rescored after taking the visual evidence into account. The hypothesis that has the highest combined score is then selected. More details about this approach can be found in [228].

8.3 Conclusions

This section attempted to give an overview of the state-of-art in the area of audiovisual automatic speech recognition. Currently, AV-ASR seems to be one of the most promising approaches in the effort to make automatic speech recognition a ubiquitous HCI technology.

However, AV-ASR technology is not yet mature enough to be deployed in commercial ASR applications. Further progress needs to be done in areas like the robust design of visual front-ends and the optimal fusion of multimodal features. In order to successfully address challenges in these areas, current methods utilized for audio-only speech recognition are probably not adequate and researchers will need to devise novel techniques, specifically tailored for multimodal problems. Interdisciplinary research initiatives can play an important role towards this goal. Additionally, the wide availability of big audiovisual speech corpora is expected to boost AV-ASR research, the same way audio-only ASR technology vastly improved after audio speech corpora where created.

In conclusion, audiovisual automatic speech recognition seems to be a very promising research area, with a great potential for significant impact in the design of pervasive HCI systems.

9 Adaptive Interfaces

9.1 Introduction

Today's computer user interfaces have advanced from command like interfaces to direct manipulation (WIMP) interfaces. Programmers and software designers usually create interfaces for pre-defined, hypothetical and prototypical users [87] and attempt to make them appropriate for a large and diverse group of users. Because of differences in the experience levels, learning/work styles, cognitive abilities etc. however, traditional interfaces often pose problems for individual users. For example, an interface might be too complex for a novice user, while appearing too simplistic to an expert user.

However, nowadays there is a significant trend towards adaptive and customizable interfaces, which use modeling and reasoning about the domain, the task, the user etc. to extract and represent user's knowledge, skills, and goals, in order to better serve them with their tasks. Such systems can for example adapt their interface to a specific user, give feedback about the user's knowledge and predict user's future behavior such as answers, goals, preferences and actions [156].

9.2 Motivation

Adaptive human computer interaction promises to support more sophisticated and natural input and output, to enable users to perform potentially complex tasks more quickly, with greater accuracy, and to improve user satisfaction. They are a promising attempt to overcome such problems resulting from increasing complexity of human-computer interaction. These systems are typically characterized by one or more of the following properties [210], [209], [208]:

- Multimodal Input
- Multimodal Output
- Interaction Management

This new class of interfaces promises *knowledge* or *agent-based dialog*, in which the interface gracefully handles errors and interruptions, and dynamically adapts to the current context and situation, needs of the task performed and the user model. This interactive approach is believed to have great potential for improving the effectiveness of human-computer interaction [187]. The overarching aim of intelligent interfaces is to both increase the interaction bandwidth between human and machine at the same time increase interaction effectiveness and naturalness by improving the quality of interaction. Effective human machine interfaces and information services will also increase access and productivity for all. Studies [310] provided empirical support for the concept that user performance can be increased when the interface characteristics match the user skill level, emphasizing the importance of adaptive user interfaces. A grand challenge of adaptive interfaces is to represent, reason, and exploit various models to more effectively process input, generate output, and manage the dialog and interaction between human and machine so that we maximize the efficiency, effectiveness, and naturalness, if not joy, of interacting.

9.3 Definitions

Literature on adaptive interfaces is very diverse. There are hundreds of articles and books that focus on narrow domains, from searching information with a Wireless Application Protocol (WAP) device to automatic task allocation systems for aircraft pilots. Because work on adaptive interfaces spans multiple disciplines, the definition of an adaptive interface varies. In the introduction of the special issue of Interacting with Computers about intelligent interface technology, Keebleq and Macredie [173] define an adaptive interface as:

> *One where the appearance, function or content of the interface can be changed by the interface (or the underlying application) itself in response to the user's interaction with it.*

Langley [187] considers an adaptive interface as a special class of learning systems and defines it as:

A software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user.

According to Rothrock et. al. [273] adaptive interfaces are defined as:

An adaptive interface autonomously adapts its displays and available actions to current goals and abilities of the user by monitoring user status, the system task, and the current situation.

9.4 The Nature of Adaptive Interfaces

One central feature of adaptive interfaces is the manner in which the system uses the learned knowledge. Some work in applied machine learning are designed to produce expert systems that is intended to replace human. However work on adaptive interfaces intends to construct *advisory-recommendation* systems, which only make recommendations to the user. These systems suggest information or generate actions that the user can always override. Ideally, the learned by the system knowledge should reflect the preferences of the individual users, thus providing personalized services to each one.

Every time the system suggests a choice to the user he/she accepts or rejects it, thus giving feedback to the system to update it's knowledge base either implicit or explicit. The system should carry out *on-line* learning, in which the knowledge base is updated each time an interaction with the user occurs. Since adaptive user interfaces collect data during their interaction with the user, one naturally expects them to improve during the interaction process, making them "learning" systems rather than "learned" systems.

Because adaptive user interfaces must learn from observing their user's behavior, another distinguishing characteristic of these systems is their need for *rapid* learning. The issue here is the number of training cases needed by the system to generate good advice. Thus, it is recommended the use of learning methods and algorithms that achieve high accuracy from small training sets over those with higher asymptotic accuracy but slower learning rates. On the other hand, the speed of interface adaptation to user's needs is desirable but not essential. An interface that learns slowly will be more useful to the user than an interface that does adapt to the user at all. However adaptive interfaces that learn rapidly will be more competitive, in the user's eyes, than ones that learn slowly.

9.5 Types of Adaptive User Interfaces and Applications

Someone can view many decision-support tasks in terms of making *recommendations*. However is such systems we have to specify certain details like how the user communicates his needs, the manner in which results are represented, or even the number of recommended items. Any specific approach in developing a recommendation system must take under consideration the above issues.

We can identify three main categories of adaptive user interfaces, whose differences have implications for the type of feedback the user must provide. The first category are the *Informative* interfaces which attempt to select or filter information for the user, presenting only those items he will find interesting or useful. The most obvious examples of such systems are for product recommendation, news filtering and Web navigation and information seeking. These systems usually directs the user's attention within a large space of items. Typical user feedback in such systems include marking recommended choices as desirable or undesirable, rating them on some scale, or giving some similar form of system's results evaluation.

9.5.1 Informative systems

Depending on the type of filtering and interaction with the user *Informative* systems, we can distinguish three main classes of them. Those that perform *content-based* filtering, those that perform *collaborative or social* filtering and those that do both. In *content-based* filtering the system presents to the user a number of recommendations and the user marks the desirable and undesirable ones. Briefly, this scheme represents each item with a set of descriptors, usually the words that occur in a document, and the filtering system uses these descriptors as predictive features when deciding whether to recommend a document to the user. Example systems of this category are [186], [58], [249]. Of course, *content-based* methods are also widely used in search engines for the World Wide Web.

In *collaborative or social* filtering the system requires the user to rate a series of sample items from which it constructs a simple profile. Then it finds other people with similar profiles to the current user and recommends items that they liked and the current user has not yet rated. Because these systems make predictions about items based on feedback on many different users the filtering is called *collaborative or social.* Example systems of *collaborative* filtering are [290] RINGO, and a number of vendors on the Web, including AMAZON.COM which sell books and other items.

Although researchers typically contrast content-based and collaborative filtering, the two approaches are not mutually exclusive. There exists systems that combine *collaborative* and *content based methods* . The intuition is that content-based methods are best for suggesting topics similar to ones the user has liked in the past, whereas collaborative methods can suggest items outside the user's normal area that he will find interesting. Example system is [44].

9.5.2 Generative systems

The second category of adaptive user interfaces are the *generative* interfaces which focus on the generation of some useful knowledge structure. Examples of this category include document preparation, drawing packages, spreadsheet programs, and systems for planning, scheduling and configuration. These areas support richer types of feedback since the user can not only override a recommendation but can replace it with another one entirely.

9.5.3 Conversational systems

The third category of adaptive user interfaces are the *conversational* interfaces. Systems with conversational interfaces, instead of accepting keywords and returning a long list of choices, they perform a dialog with the user asking a series of questions each designed to reduce the number of acceptable candidates, and the user's answers provide constraints that narrow the search. For more information see [102].

9.6 Benefits and Limitations of Adaptive Interfaces

Before an adaptive interface is built, the designer should be aware of its potential benefits and limitations.

9.6.1 Benefits

In their comparison of clumsy, nonadaptive and workload adaptive interface systems, Parasuraman, Mouloua and Hilburn [244] concluded that there are three benefits from using effective adaptive interfaces: enhanced performance, regulated workload, and reduced reliance on static automation. They showed that through use of an efficient adaptive interface, users were able to reduce the time to complete tasks and committed fewer errors while user satisfaction was increased. An adaptive user interface can minimize the need for operators to maintain or transform information in working memory when the operator's workload is high through increasing the number of tasks allocated to the machine. Therefore, if an adaptive system can reduce the workload level at peak times, it will improve the overall human-machine system performance by lowering the number of possible errors and decreasing the number of necessary personnel.

9.6.2 Limitations

Adaptive user interfaces should not be considered a panacea for all problems. The designer should seriously take under consideration if the user really needs an adaptive system. The most common concern regarding the use of adaptive interfaces is the violation of standard usability principles. In fact, there exists evidence that suggests static interface designs sometimes promote superior performance than adaptive ones. In terms of consistency, Somberg [298] found that retrieval time with a static alphabetic menu is as fast as with a frequency-based menu. From a user control perspective, Schneiderman [281] and Keeble and Macredie [173] found that users of adaptive interfaces sometimes felt that they were losing control of the system. In terms of transparency, Hook [141] and Schneiderman [281] found that adaptivity obscured the visibility of the system. Hook and Schneiderman also suggest that some adaptive systems hide the way they work and, in turn, cause a loss of user predictability.

As a consequence of these factors, an adaptive interface may engender mistrust, or even distrust. In his research for usability trade-offs in adaptive user interfaces Paymans et. al. [248] showed that adaptation mechanisms have a cost-benefit trade-off for usability. Unpredictable autonomous interface adaptation can easily reduce a system's usability. To reduce this negative effect of adaptive behavior they attempted to help users building adequate mental models for such systems. This user support attempt, improved ease of use but unexpectedly reduced learnability. This shows that an increase of ease of use can be realized without actually improving the user's mental model of adaptive systems.

9.7 Approaches to Adaptive User Interface Design

The literature on the subject of adaptive interfaces is very heterogeneous and closely linked with each domain of application. According to [273] we can classify existing designs and models along three main points of views: human-factors, human-computer interaction, and hybrid.

9.7.1 The Human Factors Approach

The human factors approach focuses on two main topics. The first topic addresses the appropriate choice of automation level and the degree to which a task must be shared between the operator and the system. The issues that are typically raised include the selection of tasks to be automated, the time at which automation should be switched on or off, and the entity (human or machine) that is responsible for the switch. The second topic focuses on the identification and measurement of the user s resources. Of particular interest is the issue of workload, which is assumed to be the main trigger of the adaptive process.

9.7.2 The Human-Computer Interaction Approach

In the HCI field, adaptive interfaces are often called Intelligent User Interfaces. This approach focuses on a table of variables that categorizes the user called the user profile. Of particular interest within the profile are user s goals and preferences. The profile is inferred by analyzing the user s behavior, which generally consists of his interactions with the system.

9.7.3 Hybrid Approaches

To incorporate the user-centered focus of the human factors approach with the systems-oriented view of the HCI approach, researchers have derived hybrid frameworks. Various approaches are taken in the design of systems using Hybrid Approaches. For more information see [157], [66], [320].

9.8 Evaluation

The literature on the subject of adaptive interfaces is very heterogeneous and closely linked with each domain. Nevertheless adaptive user interfaces differ from previous software entities in some important ways, which suggests a careful examination of the issues surrounding their experimental evaluation. In the following section we are trying to outline the most important features that could be objective measures in the evaluations of various types of adaptive user interfaces.

9.8.1 Efficiency

People typically invoke computational decision aids, including adaptive user interfaces because they expect the software will help them accomplish their tasks more rapidly and with less effort than they can do on their own. This makes *efficiency* an obvious measure to use in evaluating adaptive user interfaces. However one can instantiate this measure in different ways.

One natural measure of *efficiency* is the time the user takes to accomplish his task. But the time is not the only measure of efficiency. Another facet is the effort that the user must exert to make a decision or solve a problem. Here the most obvious measure is the number of user actions that take place during the solution of a problem.

9.8.2 Measures of Quality

Another main reason that users turn to advisory systems is to improve the *quality* of solutions to their task. This goal is very common in situation where someone wants to solve a problem that takes many steps, but is also common in situations where one wants to find an appropriate item. As with efficiency someone can define the notion of quality in different ways.

If there exists some objective measure of quality for a domain, one can use this directly in an evaluating experimental study. For example if there is a site that find the lowest price for a product the the resulting price constitutes an objective measure of quality. However, evaluating quality in domains that involve more than one criterion becomes more complicated.

9.8.3 Measures of User Satisfaction

This evaluation suggests reliance on some separate measure of user satisfaction to determine quality of the system's behavior. One way to achieve this is to present each user with a questionnaire that asks about their subjective experience.

Another measure of user's satisfaction involves giving the user some control over weather they use certain features of the system. If the user turns off the system's advisory capability or disables its personalization module, one can safely conclude that the user has not been satisfied by his experience with these features.

9.8.4 Measures of Predictive Accuracy

Because the user model in an adaptive user interface makes predictions about user responses it is natural to rely on predictive accuracy. Accuracy is the most wide spread measure in machine learning which makes it very familiar to adaptive interface developers.

However, there are some inherent problems with using predictive accuracy to determine the success of a system. Although this measure can be a useful analytical tool for understanding the details of a system behavior it does not reflect directly the overall efficiency or quality of solutions which should be the main concern.

9.9 Relation to WP10

In order to cope with an ever-increasingly large and complex Web, there is a demand for intelligent tools and structures which can simplify the experience and make navigation of the sites and information retrieval easier for users and yet maximize the quality and completeness of the experience. These tools and structures should provide sufficient intelligence so that one can sense the environment, perceive and interpret the situations in order to make decisions and to control actions.

The challenge of massive, multimedia digital libraries has turned attention toward the problem of integrated access to structured data and textual sources as well as media with spatial and temporal properties (e.g., sound, maps, images, video). Intelligent multimedia information retrieval goes beyond traditional hypertext or hypermedia environments to provide content based indexing of multiple media and management of the interaction with these materials by representing and reasoning about models of the media, user, discourse and task. Interaction tailored to the user, task, and situation will be necessary to support interaction with large scale and complex digital libraries. Heterogeneous media sources/services may require different methods of access, including distinct query languages and profiles (e.g., keywords for text data, structured query for relational data, visual query), however these should be as intuitive and uniform as possible, supporting cross media query.

10 Usability in Mobile Multimedia Applications

10.1 Introduction

Mobility poses special challenges for working with multimedia content. Mobile content should be easy to create and access with mobile devices, which have limited capabilities for input, output, storage, memory, processing power, etc. Novel methods for overcoming these difficulties are needed and this section will discuss some of them. More specifically, alternative interfaces will be covered, as well as semantics and metadata for efficient storage and retrieval of multimedia content. Some comparisons between capabilities of handheld devices and those of "traditional" PCs are also considered.

10.2 Alternative Interfaces

Traditional interfaces are not designed for mobile use. In many cases more intuitive interfaces support better mobile use. Especially with tiny mobile devices with very small display traditional WIMP-based (Windows, Icons, Menus and Pointing device) interfaces are cumbersome as only few items can be seen at one time. New smart phones equipped with camera, infrared reader and bluetooth interface allow use of more intuitive interfaces and to control with mobile device other devices.

Applications running in the device itself can be used by voice recognitions interface or by video/image-based interface.Some user interaction paradigms for physical browsing include ScanMe, PointMe and TouchMe. According to the authors these paradigms provide an optimal support for natural interaction with physical objects with any tagging technology.

User may choose the tag (object) of interest by pointing it, touching it or scanning it with his mobile device (here PDA). Tag can be RF-ID tag (Radio Frequency) or Bluetooth tag. Also visual tags are used for interaction with the physical world. [REF]

10.3 Semantic Description of Metadata

10.3.1 Ontologies

In philosophical sense the concept of ontology comes close to metaphysics and studies the nature of being. Webster dictionary defines ontology as a branch of metaphysics, concerned with the nature and relations of being. Since the term was adopted to artificial intelligence and computer science in general, it's meaning was somewhat altered. One of the most quoted definition of ontology in the scope of computer comes from Thomas Gruber, who defines the ontology as a specification of a conceptualization [125]. This transforms the concept of ontology from being an explanatory principle of the world into a design instrument applied in considering information systems.

One essential question about ontologies is the scope of one ontology. How much information and various topics should one ontology cover? At one extreme is the approach where everything should be fitted in one ontology. At the other extreme is the approach where ontologies are very small and designed again for every purpose. Both of these extremes have their drawbacks. The one-huge-ontology approach fails because it likely cannot maintain internal consistency between concepts and also, because it probably will become very slow and cumbersome to use in the technical sense. The other extreme, instead, is untenable because it dissolves the very point of using ontologies in the first place, i.e., knowledge sharing.

The most feasible way of utilizing ontologies lies somewhere in between these extreme approaches. An ontology is to enable knowledge sharing within some distinct domain. It can be envisaged, however, that using information across ontologies is evident in the future. This brings about ontology management issues. In principle ontology management deals with ontology integration. Ontology integration can be roughly divided into three categories: actual integration, merge, and use. Ontology integration is about constructing new ontologies by combining existing ontologies about different domain. In ontology merge process, existing ontologies about subjects in the same domain are harmonized to create a new ontology. Ontology use concerns the process of integrating ontologies into applications.

A special case of ontology integration comes with versioning. If an ontology is changed or replaced with a new version all of a sudden, how are the applications designed according to the previous version(s) supposed to

work? The nature of the change in the ontology has impact on this. Five different ways of changing an ontology can be identified [176]:

- Logical changes (changes in the hierarchy of concepts)
- Non-logical changes (e.g. changes in natural language documentation of concepts)
- Identifier changes (renaming concepts)
- Definitions of new concepts
- Deletion of concepts

Compatibility between different ontology versions can be divided into following categories based on the notions of prospective use and retrospective use [177]. The former refers to the case where the use of data sources that conform to a previous version of the ontology via a newer version of the ontology, and the latter to the case where the use of data sources that conform to a newer version of the ontology via a previous version of the ontology. These can be defined as:

- Fully compatible revisions (upward and backward): the semantics of the ontology is not changed, for example, syntactic changes or updates of natural language descriptions; this type of change is compatible in both prospective use and retrospective use
- Backward compatible revisions: the semantics of the ontology are changed in such a way that the interpretation of data via the new ontology is the same as when using the previous version of the ontology, for example, the addition of an independent class; this type of change is compatible in prospective use
- Upward compatible revisions: the semantics of the ontology is changed is such a way that an older version can be used to interpret newer data sources correctly, for example. the removal of an independent class; this revision is compatible in retrospective use
- Incompatible revisions: the semantics of the ontology is changed in such a way that the interpretation of old data sources is invalid, for example, changing the place in the hierarchy of a class; this type of change is incompatible in both prospective use and retrospective use

10.3.2 Description Languages

Metadata is intended for expressing the meaning or otherwise characterizing some data. This poses challenges for the languages used for expressing metadata. A metadata language should have expressive power for describing the entities within the data, as well as their interrelations. For example, should the data depict Ian Rankin the author of detective stories, the metadata language should be able to express that he is an author, that he writes detective stories, etc. Naturally the nature of data has impact on the metadata. If the depiction of Ian Rankin is a picture, possible metadata can be the colour of his hair and other features evident in the picture. If the depiction is a piece of text, instead, the colour of his hair might well be absent.

Another thing having influence on metadata is its usage purposes. If the metadata characterizing Ian Rankin is intended for listing crime writers, relevant metadata can be books, their years of publication, their main characters, etc. However, as Ian Rankin is known to have other occupations in addition to writing— grape-picker, swineherd, tax inspector, and punk musician among others—he can be categorized based on many other areas as well. This phenomenon is closely related to the scope of ontologies considered above. Metadata can be considered as an ontology to data. The inevitable domain-specifity of ontologies comes to restrict metadata as well. Metadata could—and should—not try and capture everything there is to say about a piece of data. Instead, it should be acknowledged that metadata is designed for some purpose.

XML (eXtensible Markup Language) is becoming a de facto standard for exchanging any information over the Web, and metadata is no exception. However, as such XML is too inexpressive and further specifications on top of it are needed. Some languages for expressing multimedia-related metadata are considered below. Here, instead, general metadata description languages are briefly presented. The examination is restricted to Semantic Web languages, since they are at the moment the most influental public generic metadata specification efforts.

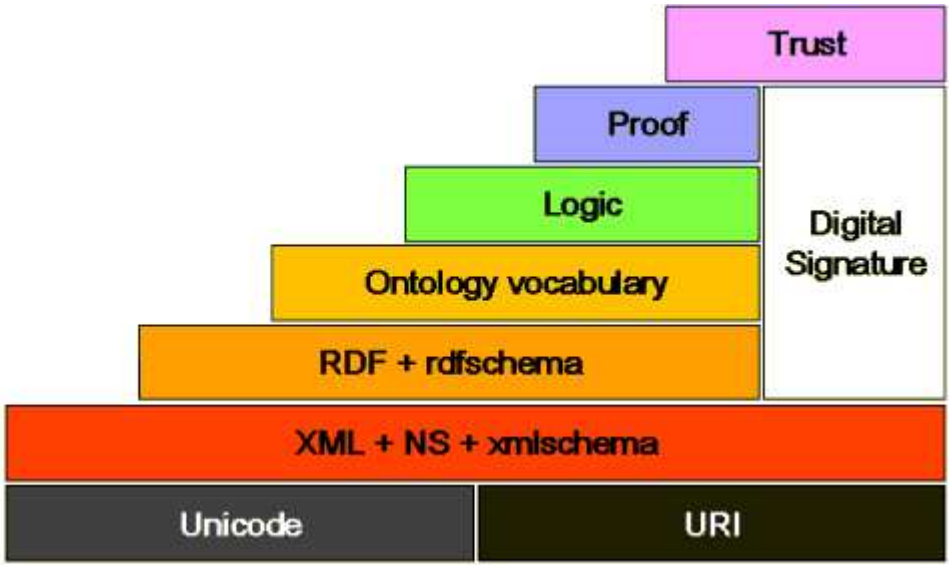

Figure 4: The Semantic Web tower

The Semantic Web is an extension to the current Web, in which the content is given machine-understandable definitions [52]. This machine-understandability means, that computer programs in addition to human beings are capable of processing material in the Web, reasoning upon it, and finally taking actions. The Semantic Web abstract architecture is often depicted as a "tower" or a "layer cake" [51]. In that architecture, depicted also in Figure 4, the lower layers act as enablers for upper layers. For example, the ontologies on the ontology layer are defined using RDF. RDF, instead, has a serialization in XML. It should be noted, however, that the upper layers are not inherently dependent on lower layers. RDF, for example, can have other serialization syntaxes besides XML. Actually, a simpler serializations for RDF such as N3 and N-Triples are popular among researchers because they more space-saving and faster to write by hand than the XML serialization of RDF.

The layers in the Semantic Web tower that are below RDF are not Semantic Web specific but used elsewhere as well. RDF (Resource Description Framework) defines the core data model of the Semantic Web, which is a directed graph [199]. That differentiates RDF from plain XML, which is an arbitrary tree. The layers on top of RDF reuse its data model. The basic RDF graph consists of subject, predicate, and object (also often called property) [178].

RDF Schema [65] specifies some basic concepts that are useful in expressing any metadata. Such are for example domain and range restrictions for properties, and subclass-relationship between classes. RDF Schema actually enables defining simple ontologies. However, the Web Ontology Language (OWL) specified on top of it has more expressivity [48]. With OWL one can specify for example cardinalities, intersections of classes, unions, etc. RDF, RDF Schema, and OWL are all since February 2004 W3C Recommendations.

There existed ontology description languages well before the Semantic Web. Examples are OCML[1], Loom, and FLogic[2]. In many ways these "traditional" ontology description languages are more expressive than the Web-based ontology languages, and therefore one could argue that they are better suitable in describing metadata. However, since Semantic Web languages are based on XML, a de facto standard and widely used, they are probably more suitable in knowledge sharing and information exchange over the Web. For comparison between various ontology description languages, see [122].

[1] Operational Conceptual Modeling Language
[2] Frame Logic

10.4 Metadata Standards for Multimedia

10.4.1 MPEG7 and MPEG21

There is huge amount of multimedia data available these days, in digital archives, on the web, personal databases etc. and the amount of the data is increasing all the time. Yet the value of that information depends on how easily it can be managed, stored and retrieved.

Multimedia Content Description Interface better known as MPEG-7 is a multimedia standard that aims to multimedia content management. It is developed by Moving Pictures Experts Group (MPEG) committee of Organization for Standardization (ISO).

MPEG-7 does not aim to any specific application, rather MPEG-7 support as broad a range of applications as possible. And it aims to be generic, which is key difference between MPEG-7 and other multimedia standards.

Four basic elements of MPEG-7 are: Descriptors, Description Schemes: Description Definition Language (DDL) and System Tools.

Descriptors: "A representation of Feature. A Descriptor defines the syntax and semantics of the Feature representation."

Description Scheme: "The structure and semantics of the relationships between its components, which may be both Descriptors and Description Schemes."

Description Definition Language: "A language that allows the creation of new Description Schemes and, possibly, Descriptors. It also allows the extension and modification of existing Description Schemes."

System Tools: "Tools to support multiplexing of descriptors with content, delivery mechanisms, and coded representations (both textual and binary formats) for efficient storage and transmission and management and protection of intellectual property in MPEG-7 Descriptors."

The MPEG-7 standard itself consist of 7 parts: MPEG-7 Systems, MPEG-7 DDL, MPEG-7 Visual, MPEG-7 Audio, MPEG-7 Multimedia Description Schemes, MPEG-7 Reference Software, MPEG-7 Conformance, MPEG-7 Extraction and Use of Descriptions.

MPEG-standards (MPEG-1,-2,-4,-7) provide a complete, powerful, and successful set of tools for multimedia representation, together with other multimedia standards like JPEG and JPEG 2000. But as these standards are constructed one by one, they do not fit perfectly together. Some parts of them overlap and on the other hand, there exists caps between them. And there is also need for integrated and suitably normative digital rights management (DRM) system. MPEG-21 aims to multimedia framework that solves these problems.

In briefly MPEG-21 is an open standards-based framework for multimedia delivery and consumption. It aims to enable the use of multimedia resources across a wide range of networks and devices.

More about MPEG-standards can be found for example at the web address http://www.chiariglione.org/mpeg/ and at http://www.mpeg.org/MPEG/starting-points.html.

10.5 Content Description, Creation, Annotation, and Sharing with Mobile Devices

Due to their limited input capabilities and other restrictions, mobile devices pose special challenges for working with multimedia content. Typically the reason for working on multimedia content with a mobile device in the first place is that the content is created with that device. It can be for example a photograph taken with a mobile phone's digital camera. Once the content is created, the user would benefit from easy-to-use functionalities storing, annotating, and possibly even editing the content.

In this context multimedia content can be considered as data and the descriptions or annotations attached to it as metadata. The relationship between multimedia content and its metadata can be investigated. A practical question is, for example, whether the metadata descriptions should be embedded within the data (i.e., in the same file), or rather stored in separate files and connected with links? And if in separate files, should the links be navigable in both directions, i.e., from metadata to data and vice versa? The Annotea project[3] represents the "separated files" approach. Annotea investigates bookmark sharing in the Web so that independent bookmark servers store the bookmarks—such as comments or recommendations based on various Web content [179]. Whenever people with access rights enter bookmarked Web content, they get notified about

[3] http://www.w3.org/2001/Annotea/

the bookmarks. So far only W3C's own Web browser/editor, called Amaya[4], supports the creation and browsing of these annotations. Storing content and metadata separately suits mobile applications, since mobile devices often have limited storage capacities. Metadata could be stored in the mobile device and the actual content in some server with more storage space.

When working with mobile devices with limited input modes, the content annotation should be automated as far as possible. Examples of relevant metadata which can be automatically annotated are time of creation and/or storage, creator, location (in case the mobile device is capable of positioning itself), type of content (PNG, SVG, MPEG, HTML, etc.), size of the content (in bytes), dimensions (in case of images), duration (in case of audio or video), resolution and color scale (in case of images and video). Based on the metadata created automatically and/or manually by the user, the system can also try to guess the actual object of the content by comparing it with other existing material and asking verification from the user [327, 278].

Mobile content sharing is an interesting and emerging phenomenon. Web logging, or "blogging", is already hugely popular and mobile devices equipped with digital cameras add a new twist to blogging. You can take pictures or shoot film anywhere, and store it to a database, which is possibly shared with other persons. There already exist some commercial initiatives for this[5]. Adding metadata to these shared mobile content would be of use, especially if the metadata would be defined according to semantics shared by the community accessing the shared databases.

10.6 Novel Devices and Content Adaptation

10.6.1 Challenges with Devices with Limited Capabilities

Mobile multimedia devices (smart phones and PDAs) have limited computing capacities. Typical clock frequency for a phone is 100 MHz and for PDA few hundreds MHz, which is quite less compared to PC's few GHz. Phones and PDAs have also small memory typical RAM is 32-62 MBytes (+ additional memory card of few hundreds MBytes), whereas PCs can have few GBytes.

Also display capacity is limited compared with PC displays. Especially with mobile phones small display is clearly a weakness concerning multimedia applications. Typical display in mobile phone varies from 176x208 to 200x 640 pixels or similar with 65000 colors and PDAs often have 240x320 pixels display.

Although mobile devices are evolving all the time, it's not in range of vision that their computing capacity would be near to that of PC's.

Image processing and pattern recognition are an old disciplines and many tasks are trivial when there is enough computing capacity. But mobile devices still have quite small memory and low computing capacity compared with PCs. Pattern recognition, image processing, editing multimedia and not to mention video editing, is still challenging task with them. New methods and algorithm need to be invented or know algorithms need to be optimized for target platform.

[4] http://www.w3.org/Amaya/

[5] See, for example, Futurice (http://www.futurice.fi); Kodak Mobile (http://www.kodakmobile.com); Cognima (http://www.cognima.com); Six Apart Typepad (http://www.typepad.com); Textamerica (http://www.textamerica.com); Nokia Lifeblog (http://www.nokia.com/lifeblog)

References

[1] http://www.nuance.com.

[2] WebSphere Voice Server(http://www-3.ibm.com/software/speech/enterprise/ep_11.html.

[3] http://studio.tellme.com.

[4] http://www.voicegenie.com.

[5] http://www.heyanita.com.

[6] http://www.speech.cs.cmu.edu/openvxi.

[7] http://www.a-s-l.com/.

[8] http://www.smarteye.se/.

[9] http://www.almaden.ibm.com/cs/blueeyes/.

[10] http://www.arringtonresearch.com/.

[11] http://www.eyelinkinfo.com/.

[12] http://www.eyegaze.com/.

[13] http://www.crsltd.com/.

[14] "IBM X+V site." http://www-306.ibm.com/software/pervasive/multimodal/.

[15] "JavaSoft, JavaBeans. Sun Microsystems, JavaBeans V1.0, 1996, http://java.sun.com/beans.."

[16] "NeuronData. Open Interface. 156 University Ave. Palo Alto, CA 94301. (415) 321-4488."

[17] "SALT forum." http://www.saltforum.org/.

[18] "Seesoft: A tool fro visualizing line-oriented software statistics." *IEEE Transactions on Software Engineering*, 18, 11 (1992) 957-968.

[19] "Visix Software Inc., Galaxy Application Environment., 1997. (Company dissolved in 1998. Galaxy was bought by Ambiencia Information Systems, Inc., Campinas, Brazil, support@ambiencia.com. http://www.ambiencia.com)."

[20] "VoiceXML 2.0 W3C Recommendation." http://www.w3.org/TR/voicexml20/.

[21] "VoiceXML forum." http://www.voicexml.org/.

[22] "W3C Composite Capability/Preference Profiles Working Group." http://www.w3.org/Mobile/CCPP/.

[23] "W3C Extensible MultiModal Annotation markup language (EMMA)." http://www.w3.org/TR/emma/.

[24] "W3C Ink Markup Language." http://www.w3.org/TR/InkML/.

[25] "W3C MultiModal Interaction Working Group." http://www.w3.org/2002/mmi/.

[26] "W3C MultiModal Interaction Working Group : Multimodal Interaction Framework." http://www.w3.org/ TR/mmi-framework/.

[27] "W3C MultiModal Interaction Working Group : System and Environment Framework." http://www.w3.org/TR/sysenv/.

[28] "W3C Voice Browser Activity." http://www.w3.org/voice/.

[29] "W3C Web Accessibility Initiative." http://www.w3.org/WAI/.

[30] "XVT Software, Inc. XVT. Box 18750 Boulder, CO 80308. (303) 443-4223."

[31] Martin G. Helander, " Design of Visual Displays.*Handbook of Human Factors 1987* n.5.1 p.507-548 New York John Wiley & Sons."

[32] Acero, A. and Stern, R. M. (1990), "Environmental robustness in automatic speech recognition." In Proceedings of the 1990 International Conference on Acoustics, Speech, and Signal Processing, pages 849-852, Albuquerque, New Mexico.Institute of Electrical and Electronic Engineers., 1990.

[33] A. Adjoudani and C. Benoît, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by Humans and Machines* (D. Stork and M. Hennecke, eds.), pp. 461–471, Berlin, Germany: Springer, 1996.

[34] Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., and Traum, D. R. (1995), "The trains project: a case study in building a conversational planning agent.." Journal of Experimental and Theoretical AI., 1995.

[35] Allen, J., Hunnicutt, M.S. and Klatt, D. (1987), "From text to speech the mitalk system.." MIT Press, Cambridge, Massachusetts., 1987.

[36] Alshawi, H., Arnold, D. J., Backofen, R., Carter, D. M., Lindop, J., Netter, K., Pulman, S. G., and Tsujii, J.-I. (1991), "Rule formalism and virutal machine design study.." Technical Report ET6/1, CEC., 1991.

[37] Alshawi, H., editor (1992), "The core language engine.." MIT Press, Cambridge, Massachusetts., 1992.

[38] André, E., Rist, T., "Presenting Through Performing: On the Use of Multiple Lifelike Characters in Knowledge-Based Presentation Systems." In Proceedings of the Second International Conference on Intelligent User Interfaces (IUI 2000): 1-8, 2000.

[39] Andrews, Keith, "Visualizing cyberspace: Information visualization in the harmony interner browser." *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Amalitos,CA(1995), 97-104.

[40] G. Antoniol, R. Fiutem, G. Lazzari, and R. De Mori, "System architecture and applications," in *Spoken Dialogues with Computers* (R. De Mori, ed.), (Academic Press), pp. 583–621, 1997.

[41] Applebaum, T. H. and Hanson, B. A., "Regression features for recognition of speech in quiet and in noise.." In Proceedings of the 1989 International Conference on Acoustics, Speech, and Signal Processing, pages 985, Glasgow, Scotland. Institute of Electrical and Electronic Engineers., (1989).

[42] Asahi, T., Turo, D., and Shneiderman, B., "Using treemaps to visualize the analytic hierarchy process." *Information Systems Research*, 6, 4 (December 1995), 357-375.

[43] B. Pan, H. Hembrooke, G. Gay, L. Granka, M. Feusner, and J. Newman , "The determinants of web page viewing behavior: An eye tracking study." In S.N. Spencer (Ed.), Proceedings of Eye Tracking Research & Applications, New York: ACM SIGGRAPH.

[44] Balabanovic, M., "Exploring versus exploiting when learning user models for text recommendation." User Modeling and User-Adapted Interaction 8: pp. 71-102, 1998.

[45] Bartkova, K. and Sorin, C. (1987)., "A model of segmental duration for speech synthesis in french.." Speech Communication, 6:245-260., 1987.

[46] Bates, M., Bobrow, R., Ingria, R., and Stallard, D. (1994)., "The delphi natural language understanding system.." In Proceedings of the Fourth Conference on Applied Natural Language Processing, pages 132-137, Stuttgart, Germany. ACL, Morgan Kaufmann., 1994.

[47] M. J. Beal, N. Jojic, and H. Attias, "A graphical model for audio-visual object tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 828–836, July 2003.

[48] S. Bechhofer *et al.*, "OWL Web Ontology Language Reference," World Wide Web Consortium, Feb. 2004. W3C Recommendation, available at: http://www.w3.org/TR/owl-ref/.

[49] H. Beigi, "An overview of handwriting recognition," 1993.

[50] Bengio, Y., DeMori, R., Flammia, G., and Kompe, R. (1992)., "Global optimization of a neural network—hidden markov model hybrid.." IEEE Transactions on Neural Networks, 3(2):252-259., 1992.

[51] T. Berners-Lee, "Semantic web on XML, XML 2000 conference," December 2000. Available at: http://www.w3.org.

[52] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.

[53] N. Bernsen and L. Dybkjoer, "Is speech the right thing for your application?," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[54] Bhaskar T. N., Foo Tun Keat, Ranganath Surendra, Venkatesh Y. V. (2003), "Blink detection and eye tracking for eye localization." IE EE Tencon, India., 2003.

[55] Bocchieri, E. L. (1993)., "Vector quantization for the e cient computation of continuous density likelihoods.." In Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 692-694, Minneapolis, Minnesota. Institute of Electrical and Electronic Engineers., 1993.

[56] Bolt, R., "Put-That-There : Voice and gesture at the graphics interface." Computer Graphics, 14(3): 262-270, 1980.

[57] Bolt, RA. 1982, "Eyes at the interface." Proc. ACM CHI'82, 360-362, 1982.

[58] Boone, G., "Concept features in re:agent, an intelligent email agent." Proceedings of the Second International Conference on Autonomous Agents, pp. 141-148. Minneapolis, MN: ACM Press, 1998.

[59] Borning A., "The Programming Language Aspects of Thinglab; a Constraint-Oriented Simulation Laboratory. ACM Transactions on Programming Languages and Systems, 1981. 3(4) pp. 353-387."

[60] Bouma, G., Koenig, E., and Uszkoreit, H. (1988)., "A flexible graph-unifcation formalism and its application to natural-language processing.." IBM Journal of Research and Development., 1988.

[61] Brad A. Myers, Dario A. Giuse, Roger B. Dannenberg, Brad Vander Zanden, David S. Kosbie, Edward Pervin, Andrew Mickish, and Philippe Marchal., "Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces. IEEE Computer 23, 11 Nov. 1990, 71-85."

[62] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 994–999, 1997.

[63] C. Bregler and Y. Konig, "Eigenlips for robust speech recognition," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 669–672, 1994.

[64] Bresnan, J., editor (1982), "The mental representation of grammatical relations.." MIT Press, Cambridge, Massachusetts, 1982.

[65] D. Brickley *et al.*, "RDF Vocabulary Description Language 1.0: RDF Schema," World Wide Web Consortium, Feb. 2004. W3C Recommendation, available at: http://www.w3.org/TR/rdf-schema/.

[66] Brusilovsky, P., "Methods and techniques of adaptive hypermedia." In Journal of user modeling and user-adapted interaction.6, 2-3 (1996), pp. 87-129, 1996.

[67] Bush V, "As We May Think. The Atlantic Monthly, p. 101-108; July, 1945."

[68] Buxton and William, "There's more to interaction than meets the eye: Some issues inmanual input. In Norman, D.A., and Draper, S.W. (Editors), *User Centered System Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ (1985) 319-337."

[69] Buxton W. et al, "Towards a Comprehensive User Interface Management System, in Proceedings SIGGRAPH'83: Computer Graphics. 1983. Detroit, Mich:17. pp. 35-42."

[70] Cakir A. and Stewart T.F.M., "The VDT Manual *John Wilew and Sons*, New York (1980) ."

[71] Campbell, W. N. (1992)., "Syllable-based segmental duration.." In Bailly, G. and Benoit, C., editors, Talking Machines: Theories, Models, and Designs, pages 211-224. Elsevier Science., 1992.

[72] Card, Stuart K., Machinlay, Jock D., and R. George G., "A morphological analysis of the design space of input devices, *ACM Transactions on Information Systems*, 9,2 (1991), 99-122."

[73] Card S.K., Moran T.P. and Newell A., "The psychology of Human-Computer Interaction." Hillsdale, NJ:Erlbaum.

[74] Card, Stuart K., Robertson, George G., and York, William, "The webbook and the webforager: An information workspace for the world-wide-web." *Proc. CHI'96 Conference: Human Factors in Computing Systems*, ACM, New York (1996).

[75] Carlson, R. and Granstrom, B. (1986)., "A search for durational rules in a realspeech data base.." Phonetica, 43:140-154., 1986.

[76] Carpenter, B. (1992)., "The logic of typed feature structures,." volume 32 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press., 1992.

[77] Carpenter, R., "The logic of typed feature structures." Cambridge University Press, 1992.

[78] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int'l Journal of Comp. Vision*, vol. 22, pp. 61–79, February 1997.

[79] Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjlmsson, H. & Yan. H., "Embodiment in conversational interfaces: Rea." Proceedings of the Association for Computing Machinery (ACM) Special Interest Group on Computer Human Interaction (SIGCHI), Pittsburgh, PA., May 1999, pp. 520-527, 1999.

[80] D. Chandramohan and P. L. Silsbee, "A multiple deformable template approach for visual speech recognition," in *Proc. Int'l Conf. on Spoken Language Processing*, pp. 50–53, 1996.

[81] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proceedings of the IEEE*, vol. 92, pp. 485–494, Mar. 2004.

[82] Chimera, Richard, "Value bars: An information visualization an navigation tool for multiattribute listings." *Proc. CHI'92 Conference: Human Factors in Computing Systems*, ACM, New York (1992), 293-294, 1992.

[83] G. Chiou and J.-N. Hwang, "Lipreading from color video," *IEEE Trans. on Image Processing*, vol. 6, pp. 1192–1195, Aug. 1997.

[84] Christopher Ahlberg and Ben Shneiderman , "Visual information seeking: Tight coupling of dynamic query filters with starfield displays." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems* 1994 v.1 p.313-317.

[85] Chuah Mei C., Roth, Steven F., Mattis, Joe, and Kolojejchcik, John, "Sdm: Malleable informations graphics." *Proc. IEEE Information Visualization'95*, IEEE Computer Press, Los Alamitos, CA, (1995), 66-73.

[86] Cohen, P., Oviatt, S., "The Role of Voice in Human-Machine Communication." In Voice Communication Between Humans and Machines. Roe, D., Wilpon, J. (editors). National Academy Press, Washington D.C.: 34-75, 1994.

[87] Cooper, A., "The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity." IN: SAMS Macmillan Computer Publishing, 2000.

[88] T. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. Europ. Conf. on Comp. Vision*, vol. II, pp. 484–498, Springer-Verlag, 1998.

[89] T. F. Cootes, T. C.J., C. D. H., and J. Graham, "Active shape models - their training and application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[90] Corno F., Farinetti L. and Signorile I. (2002), "A cost effective solution for eye-gaze assistive technology." IEEE Int. Conf. on Multimedia and Expo, August 26-29, Lausanne, 2002.

[91] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L.-W. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg, "Distributed meetings: A meeting capture and broadcasting system," in *ACM Conf.Multimedia*, pp. 123–132, 2002.

[92] R. De Mori, "Problems and methods for solution," in *Spoken Dialogues with Computers* (R. De Mori, ed.), (Academic Press), pp. 1–22, 1997.

[93] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Artificial Intelligence*, vol. 93, no. 1-2, pp. 1–27, 1989.

[94] Digalakis, V. and Murveit, H.(1994), "Genones: Optimizing the degree of mixture tying in a large vocabulary hidden markov model based speech recognizer.." In Proceedings of the 1994 International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 537-540, Adelaide, Australia. Institute of Electrical and Electronic Engineers., 1994.

[95] Dix A., Finlay J., Abowd G., Beale R., "Human-Computer Interaction." Prentice Hall, 1993.

[96] E. Dougherty and C. Giardina, *Image Processing - Continuous to Discrete*, vol. I: Geometric, Transform, and Statistical Methods. Prentice Hall, 1987.

[97] Duchowski, A. T., Medlin, E., Cournia, N., Murphy, H., Gramopadhye, A., Nair, S., Vorah, J., Melloy, B. (2002), "3d eye movement analysis." Behavior Research Methods, Instruments, & Computers (BRMIC), 34(4), pp.573-591, 2002.

[98] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* Wiley, 2nd ed., 2000.

[99] R. Eberts, *User Interface Design.* Englewood Cliffs, NJ: Prentice–Hall, 1994.

[100] Edwards G. (1998), "A tool for creating eye-aware applications that adapt to changes in user behaviors." International ACM Conference on Assistive Technologies: 67-74, 1998.

[101] Egenhofer, Max and Richards J., "Exploratory access to geographical data based on the map-overlay metaphor." *Journal of Visual Languages and Computing*, 4, 2 (1993).

[102] Elio, R., and Haddadi, A., "Dialog management for an adaptive database assistant." (Technical Report 98-3). Daimler-Benz Research and Technology Center, Palo Alto, CA, 1998.

[103] Elting, C., Rapp, S., Mohler, G., Strube, M., "Architecture and Implementation of Multimodal Plug and Play." ICMI 03, November 5 7, 2003, Vancouver, British Columbia, Canada, 2003.

[104] Emele, M. and Zajac, R. (1990), "Typed unification grammars.." In Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania. Association for Computational Linguistics., 1990.

[105] Engelbart Douglas C., and English, William K., "A Research Center for Augmenting Human Intellect, in AFIPS Conference Proceedings of the 1968 Fall Joint Computer Conference 33 (San Francisco CA, December 1968), 395-410."

[106] Etienne Grandjean and Gavriel Salvendy, "Design of VDT Workstations. Human Factors in the Design and Use of Computing Systems (1987) n.11.1 p.1359-1397 New York John Wiley & Sons ."

[107] Farid M., Murtagh F. and Starck J.L. (2002), "Computer display control and interaction using eye-gaze." Journal of the Society for Information Display, Vol 10, No 3, pp 289-29., 2002.

[108] Fishkin, Ken and Stone, Maureen C., "Enhanced dynamic queries via movable filters." *Proc CHI'95 Conference:Human Factors in Computing Systems*, ACM,New York (1995), 415-420.

[109] Flippo, F., Krebs, A., Marsic, I., "A Framework for Rapid Development of Multimodal Interfaces." ICMI 03, November 5 7, 2003, Vancouver, British Columbia, Canada, 2003.

[110] Foley, James D., Van Dam, Andries, Feiner, StevenK., Hughes, and John F., "*Computer Graphics: Principles and Practice* (Second Edition),Addison-Wesley, Reading, MA (1990)."

[111] Foley, James D., Wallace, Victor L., Chan, and Peggy, "The human factors of computer graphics interaction techniques, *IEEE Computer Graphics and Applications*, 4, 11 (November 1984), 13-48."

[112] Fujisaki, T., Jelinek, F., Cocke, J., Black, E., and Nishino, T. (1989)., "A probabilistic parsing method for sentence disambiguation.." In Proceedings of the International Workshop on Parsing Technologies, Pittsburgh., 1989.

[113] Furui, S. (1986b)., "Speaker-independent isolated word recognition using dynamic features of the speech spectrum.." IEEE Transactions on Acoustics, Speech and SignalProcessing, 29(1):59-67, 1986.

[114] Gales, M. J. F. and Young, S. J. (1992)., "An improved approach to the hidden markov model decomposition of speech and noise.." In Proceedings of the 1992 International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 233-236, San Francisco.Institute of Electrical and Electronic Engineers., 1992.

[115] Galitz, W.O. (1992), "User-interface Screen Sesign. Wellesley, MA:QED Publishing Group."

[116] Gauvain, J.-L. and Lee, C.-H. (1991), "Bayesian learning for hidden markov model with gaussian mixture state observation densities.." In Eurospeech '91, Proceedings of the Second European Conference on Speech Communication and Technology, pages 939-942, Genova, Italy. European Speech Communication Association., 1991.

[117] George W. Fitzmaurice, "Situated information spaces and spatially aware palmtop computers, *Communications of the ACM* Volume 36 , Issue 7 (July 1993) Special issue on computer augmented environments: back to the real world, Pages: 39 - 49."

[118] George W. Fitzmaurice, Hiroshi Ishii, and William Buxton, "Laying the Foundations for Graspable User Interfaces Papers: Innovative Interaction II *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1995 v.1 p.442-449."

[119] Z. Ghahramani and M. Jordan, "Factorial hidden markov models," in *Proc. Advances in Neural Information Processing Systems*, vol. 8, pp. 472–478, 1995.

[120] Golberg, A., ed., "A History of Personal Workstations . 1988, Addison-Wesley Publishing Company: New York, NY. 537."

[121] Goldstein, Jade and Roth, StevenF., "Using aggregation and dynamic queries for exploring large data sets." *Proc. CHI'95 Conference: Human Factors in Computing Systems*, ACM,New York (1995), 23-29.

[122] O. Gorcho and A. Gomez-Perez, "Ontology evaluating knowledge representation and reasoning capabilities of ontology specification languages," in *Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods*, (Berlin, Germany), 2000.

[123] Green M., "The University of Alberta User Interface Management System, in Proceedings SIGGRAPH'85: Computer Graphics. 1985. San Francisco, CA: 19. pp. 205-213."

[124] Greenstein, Joel, Arnaut, and Lynn, "Input devices. in helander, martin, *hanbook of human-computer interaction*, north-holland, amsterdam, the netherlands (1988), 419-516."

[125] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[126] Gustafson, J., *Developing Multimodal Spoken Dialogue Systems. Empirical Studies of Human-Computer Interaction.* PhD thesis, Departmnet of Speech, Music and Hearing, KTH, 2002.

[127] Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granstrm, B., House, D., Wirn, M., "AdApt - a multimodal onversational dialogue system in an apartment domain." In Proceedings of 6th International Conference on Spoken Language Processing (ICSLP 2000): 134-137, 2000.

[128] Gustafson, J., Lindberg, N., Lundeberg, M., "Experiences from the development of August - a multimodal spoken dialogue system." In Proceedings of the ESCA tutorial and research workshop on Interactive Dialogue in Multi-Modal Systems, IDS 99, 1999.

[129] Haddock, J. N., Klein, E., and Morrill, G. (1987), "Unification categorial grammar, unification grammar and parsing." University of Edinburgh., 1987.

[130] P. Hallinan, G. Gordon, A. Yuille, P. Giblin, and D. Mumford, *Two– and Three–dimensional Patterns of the Face.* A. K. Peters, Ltd., 1999.

[131] Hansen J.P., A.W. Anderson, and P. Roed (1995), "Eye gaze control of multimedia systems." Symbiosis of Human and Artifact Y. Anzai, K. Ogawa, and H. Mori (eds), Vol 20A, Elsevier Science, pp 37-42, 1995.

[132] Henderson Jr D.A., "The Trillium User Interface Design Environment, in Proceedings SIGCHI'86: Human Factors in Computing Systems. 1986. Boston, MA:pp. 221-227."

[133] Hendley, R.J., Drew,N.N., Wood, A.S., "Narcissus: Visualizing information." *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Amalitos,CA(1995), 90-96.

[134] M. Hennecke, D. Stork, and K. Prasad, "Visionary speech: Looking ahead to practical speechreading systems," in *Speechreading by Humans and Machines* (D. Stork and M. Hennecke, eds.), pp. 331–349, Berlin, Germany: Springer, 1996.

[135] Hermansky, H., Morgan, N., and Hirsch, H. G. (1993)., "Recognition of speech in additive and convolutional noise based on rasta spectral processing.." In Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 83-86, Minneapolis, Minnesota. Institute of Electrical and Electronic Engineers., 1993.

[136] Hermansky, H., Morgan, N., Bayya, A., and Kohn, P. (1991), "Compensation for the effects of the communication channel in auditory-like analysis of speech.." In Eurospeech '91, Proceedings of the Second European Conference on Speech Communication and Technology, pages 1367-1370, Genova, Italy. European Speech Communication Association., 1991.

[137] J. Hershey and M. Case, "Audio-visual speech separation using hidden markov models," in *Proc. Advances in Neural Information Processing Systems*, vol. 14, 2002.

[138] Hill R.D., et al., "The Rendezvous Architecture and Language for Constructing Multiuser Applications. ACM Transactions on Computer-Human Interaction, 1994. 1(2) pp. 81-125."

[139] Hirsch, H. G., Meyer, P., and Ruehl, H. W. (1991)., "Improved speech recognition using high-pass ltering of subband envelopes.." In Eurospeech '91, Proceedings of the Second European Conference on Speech Communication and Technology, pages 413-416, Genova, Italy. European Speech Communication Association., 1991.

[140] L. Hirschman et al, "Multi-site data collection and evaluation in spoken language understanding," in *Proc. of the Human Language Technology Workshop*, Mar. 1993.

[141] Hook, K., "Designing and evaluating intelligent user interfaces." In Proceedings of the IUI 99. ACM press, 1999.

[142] Huang, X. D. and Lee, K. F. (1993)., "On speaker-independent, speakerdependent, and speaker-adaptive speech recognition.." IEEE Transactions on Speech and Audio Processing, 1(2):150-157., 1993.

[143] Huang, X. D., Ariki, Y., and Jack, M. (1990), "Hidden markov models for speech recognition." Edinburgh University Press.

[144] Hudson S. and King R., " A Generator of Direct Manipulation Office Systems. ACM Trans. on Office Infomation Systems, 1986. 4(2) pp. 132-163."

[145] Hudson S.E. and Smith I., "Ultra-Lightweight Constraints, in Proceedings UIST'96: ACM SIGGRAPH Symposium on User Interface Software and Technology.1996. Seattle, WA: pp. 147-155 http://www.cc.gatech.edu/gvu/ui/sub_arctic/."

[146] Hunt, M. J. and Lef ebvre, C. (1989)., "A comparison of several acoustic representations for speech recognition with degraded and undegraded speech.." In Proceedings of the 1989 International Conference on Acoustics, Speech, and Signal Processing, pages 262-265, Glasgow, Scotland. Institute of Electrical and Electronic Engineers., 1989.

[147] International Business Machines (IBM), "Cost Justifying Ease of Use. http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/23 ."

[148] International Business Machines (IBM), "*Systems Applicaton Architecture Common User Access Advanced Interface Design Reference* (SC34-4289)."

[149] Isabelle Guyon and Colin Warwick, "Handwriting as computer interface."

[150] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Europ. Conf. on Comp. Vision*, vol. I, pp. 343–356, 1996.

[151] M. Isard and A. Blake, *Snakes: Active Contour Models.* Springer, 1998.

[152] Jacob, Robert J.K., Legget, John, Myers, BradA., Pausch, and Randy, "International styles and input/output devices, *Behaviour & Information Technology*, 12, 2, (1993), 69-79."

[153] Jacob, R. J.K., "The Use of Eye Movements in HumanComputer Interaction Techniques: What You Look at is What You Get, *ACM Transactions on Information Systems*, 9, 3 (April 1991), pp 152-169."

[154] Jacob R.J.K., "A Specification Language for Direct Manipulation Interfaces. ACM Transactions on Graphics, 1986. 5(4) pp. 283-317."

[155] Jaimes A., Pelz J.B., Grabowski T., Babcock J., and Chang S.F. (2001), "Using human observers' eye movements in automatic image classifiers." Proceedings of SPIE Human Vision and Electronic Imaging VI, San Jose, CA, 2001.

[156] Jameson, A., Cecile Paris and Tasso, C., "User modeling." Proceedings of UM 97, New-York: Springer Wien New York, 1997.

[157] Jameson, A., Schafer, R., Weis, T., Berthold, A., Weyrath, T., "Making systems sensitive to the user's time and working memory constraints." In Proceedings of the IUI 99. ACM press, 1999.

[158] Jerding, Dean F. and Stasko, John T.,, "The information mural: A technique for displaying and navigating large information spaces." *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 1-10.

[159] Joel McCormack, Paul Asente, "An overview of the x toolkit." ACM SIGGRAPH Symposium on User Interface Software and Technology,Proceedings UIST 88, Banff, Alberta, Canada, Oct.,1988, pp. 46-55, 1988.

[160] John K. Ousterhout, "An X11 Toolkit Based on the Tcl Language. Winter, USENIX, 1991, pp. 105-115."

[161] Johnston, M., "Unification-based Multimodal Parsing." Proceedings of COLING-ACL, 1998.

[162] Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., and Smith, I., "Unification-based multimodal integration." In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and of the 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, 7 12 July 1997, pages 281 288, 1997.

[163] M. J. Jones and T. Poggio, "Multidimensional morphable models: A framework for representing and matching object classes," *Int'l Journal of Comp. Vision*, pp. 107–131, 1998.

[164] Joshi, A. K. and Schabes, Y. (1992)., "Tree-adjoining grammars and lexicalized grammers.." In Tree Automata and LGS. Elsevier Science, Amsterdam., 1992.

[165] Juang, B. H., Rabiner, L. R., and Wilpon, J. G. (1986)., "On the use of bandpass liftering in speech recognition.." In Proceedings of the 1986 International Conference on Acoustics, Speech, and Signal Processing, pages 765-768, Tokyo. Institute of Electrical and Electronic Engineers., 1986.

[166] Karttunen, L. (1989). Radical lexicalism. In Baltin, M. and Kroch, A., editors,, "Alternative conceptions of phrase structure.." The University of Chicago Press, Chicago., 1989.

[167] Kasik, D.J., "A User Interface Management System, in Proceedings SIGGRAPH'82: Computer Graphics, 16(3). 1982. Boston, MA: 16. pp. 99-106."

[168] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l Journal of Comp. Vision*, vol. 1, pp. 321–331, Jan. 1988.

[169] R. Kaucic and A. Blake, "Accurate, real-time, unadorned lip tracking," in *Proc. Int'l Conf. on Comp. Vision*, 1998.

[170] Kay A., "The Reactive Engine. PhD Thesis, Electrical Engineering and Computer Science University of Utah, 1969, 327."

[171] Kay, M., "Functional grammar." Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society, 142-158, 1979.

[172] Kay, M. (1984), "Functional unification grammar: a formalism for machine translation.." In Proceedings of the 10th International Conference on Computational Linguistics, Stanford University, California. ACL., 1984.

[173] Keeble R.J. and Macredie R.D., "Assistant agents for the world wide web intelligent interface design challenges." Interacting with Computers-12(4), pp. 357-381, 2000.

[174] Keim, D.A. and Kriegal, h., "Visdb: Database exploration using multidimensional cisualization." *IEEE Computer Graphics and Applications* (September 1994), 40-49.

[175] D. H. Klatt, "Review of text–to–speech conversion for english," *Journal of the Acoustical Society of America*, vol. 82, pp. 737–793, Sept. 1987.

[176] M. Klein *et al.*, "Ontology versioning and change detection on the web," in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, Lecture Notes in Computer Science, (Sigüenza, Spain), pp. 197–212, Springer, 2002.

[177] M. Klein and D. Fensel, "Ontology versioning on the semantic web," in *Proceedings of the International Semantic Web Working Symposium (SWWS)*, (Stanford, CA), June 2001.

[178] G. Klyne *et al.*, "Resource Description Framework (RDF): Concepts and Abstract Syntax," World Wide Web Consortium, Feb. 2004. W3C Recommendation, available at: http://www.w3.org/TR/rdf-concepts/.

[179] M. Koivunen *et al.*, "Annotea shared bookmarks," in *Proceedings of the KCAP 2003 workshop on knowledge markup & semantic annotation*, (Sanibel, FL), Oct. 2003.

[180] Korfhage, Robert, "To see or not to see: Is that the query?." *Communications of the ACM*, 34 (1991), 134-141.

[181] Koved L. and Shneiderman B., "Embedded menus: Selecting items in context. Communications of the ACM, 1986. 4(29) pp. 312-318."

[182] Krieger, H.-U. and Schaefer, U. (1994), "Tdl—a type description language of hpsg.." Technical report, Deutsches Forschungszentrum fur Kunstliche Intelligenz GmbH, Saarbrucken, Germany., 1994.

[183] Kumar, S., Cohen, P.R., "Towards a fault-tolerant multi-agent system architecture." Fourth International Conference on Autonomous Agents 2000, 459-466. Barcelona, Spain: ACM Press, 2000.

[184] H.-K. Kuo, A. Pargellis, and C.-H. Lee, "Information and services manager customizes dialogue-based applications," in *Proc. ESCA Workshop Interact. Dialog. Multi-Modal Syst.*, (Kloster Irsee, Germany), June 1999.

[185] L. Lamel, "Spoken language dialog system development and evaluation at LIMSI," in *Proc Internat. Symposium on Spoken Dialogue*, (Sydney, Australia), Nov. 1998.

[186] Lang, K., "Newsweeder: Learning to filter news." Proceedings of the Twelfth International Conference on Machine Learning, pp. 331-339. Lake Tahoe, CA: Morgan Kaufmann, 1995.

[187] Langley, P., "User modeling in adaptive interfaces." Proceedings of the Seventh International Conference on User Modeling (pp. 357-370), 1998.

[188] Lari, K. and Young, S. J. (1990), "The estimation of stochastic context-free grammars using the inside-outside algorithm.." Computer Speech and Language Processing, 4:35-56., 1990.

[189] Laurini, R. and Thomson, D., "*Fundamentals of Spatial Information Systems.*" Academic Press, New York (1992).

[190] Lee Stone, Brent Beutter, et al., "Models of tracking and search eye-movement behaviour." NASA.

[191] Lester, J., Converse, S.A., Stone, B.A., and Kahler, SE., "Animated Pedagogical Agents and Problem-Solving Effectiveness: A Large-Scale Empirical Evaluation."

[192] M. Leventon, W. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. I, pp. 316–323, 2000.

[193] S. Li and Z. Zhang, "Foatboost learning and statistical face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1112–1123, Sept. 2004.

[194] Lippmann, R. P., Martin, F. A., and Paul, D. B. (1987)., "Multi-style training for robust isolated-word speech recognition.." In Proceedings of the 1987 International Conference on Acoustics, Speech, and Signal Processing, pages 709-712, Dallas. Institute of Electrical and Electronic Engineers., 1987.

[195] Lockwood, P., Boudy, J., and Blanchet, M. (1992), "Non-linear spectral subtraction (nss) and hidden markov models for robust speech recognition in car noise environments.." In Proceedings of the 1992 International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 265-268, San Francisco. Institute of Electrical and Electronic Engineers., 1992.

[196] J. Luettin, *Visual Speech and Speaker Recognition.* PhD thesis, Univ. of Sheffield, May 1997.

[197] MacKenzie, I. S. and Soukoreff, R.W., "Text entry for mobile computing: Models and methods, theory and practice." Human-Computer Interaction, 17, 147–198, 2002.

[198] J. Mankoff and G. D. Abowd, "Cirrin: A word-level unistroke keyboard for pen input," in *ACM Symposium on User Interface Software and Technology*, pp. 213–214, 1998.

[199] F. Manola *et al.*, "RDF Primer," World Wide Web Consortium, Feb. 2004. W3C Recommendation, available at: http://www.w3.org/TR/rdf-primer/.

[200] K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis*. Academic Press, 1979.

[201] Mark A. Linton, John M. Vlissides and Paul R. Calder., "Composing userinterfaces with InterViews.." IEEE Computer 22, 2 Feb. 1989, 8-22, 1989.

[202] Markel, J. D. and Gray, Jr., "Linear Prediction of Speech.." Springer-Verlag, Berlin., 1976.

[203] Martin, D. L., Cheyer, A. J., Moran, D. B., "The open agent architecture: A framework for building distributed software systems." Applied Artificial Intelligence, 13, 91-128, 1999.

[204] D. Massaro and D. Stork, "Speech recognition and sensory integration," *American Scientist*, vol. 86, no. 3, pp. 236–244, 1998.

[205] T. Masui, "An efficient text input method for pen-based computers," in *Proceedings of the Conference on Human Factors in Computing Systems, CHI 98*, pp. 328–335, 1998.

[206] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 198–213, Feb. 2002.

[207] I. Matthews, G. Potamianos, C. Neti, and J. Luettin, "A comparison of model and transform-based visual features for audio-visual LVCSR," in *Proc. Int'l Conf. on Mult. and Expo*, 2001.

[208] Maybury, M. and Wahlster, W. (eds.), "Readings in intelligent user interfaces." Morgan Kaufmann Press, 1998.

[209] Maybury, M. T., "Keynote. intelligent multimedia for the new millennium." Proceedings of Eurospeech 99. Budapest, September 6-9, 1999. vol 1. p. KN1-15, 1999.

[210] Maybury, M. T. (ed.), "Intelligent multimedia interfaces." AAAI/MIT Press. 405 pp. ISBN 0-262-63150-4, 1993.

[211] Mayhew D.J. (1992), "*Principles and Guidelines in Software Interface Design.* Englewood Cliffs, NJ:Prentice Hall ."

[212] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, 1976.

[213] McTear, M., "Spoken Dialogue Technology: Enabling the Conversational Interface." ACM Computing Surveys, 34, 1, March 2002: 90-169, 2002.

[214] Microsoft Corporation (1992), "*The Windows Interface: An Application Design Guide.* Redmond,WA:Microsoft Press ."

[215] Microsoft Corporation (1995), "*The Windows Interface Guidelines for Software Design..* Redmond,WA:Microsoft Press ."

[216] Microsoft Corporation (2001), "msdn.microsoft.com.library ."

[217] Minsky, M., "A framework for representing knowledge." In P. Winston (Ed.), The Psychology of Computer Vision (pp 211-277). New York: McGraw-Hill, 1975.

[218] Mitchell, Richard, Day, Dabid, and Hirschman, Lynerre, "Fishing of information on the internet." *Proc. IEEE Information Visualization'95*, IEEE Computer Press, Los Alamitos, CA, (1995), 105-111.

[219] Montague, R. (1973), "The proper treatment of quanti cation in ordinary english.." In Hintikka, J., editor, Approaches to Natural Language, pages 221-242. Reidel., 1973.

[220] Mukherjea, Sougata, Foley, James D., and Hudson, Scott, "Visualizing complex hypermedia networks througn multiple hierarchical views." *Proc. of ACM'95 Conference: Human Factors in Computing Systems*,ACM, New York (1995), 331-337.

[221] Mulligan J. B. and Beuttler B. R. (1995), "Eye movement tracking using compressed video images." Vision Sciences and Its Applications; Optical Society Technical Digest Series, Vol.1, pp.163-166, 1995.

[222] B. Myers, S. E. Hudson, and R. Pausch, "Past, present, and future of user interface software tools," *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 1, pp. 3–28, 2000.

[223] Myers B.A. et al., "The Amulet Environment: New Models for Effective User Interface Software Development. IEEE Transactions on Software Engineering, 1997. 23(6) pp. 347-365. June."

[224] Nakajima, S. and Hamada, H. (1988)., "Automatic generation of synthesis units based on context oriented clustering.." In Proceedings of the 1988 International Conference on Acoustics, Speech, and Signal Processing, pages 659-662, New York. Institute of Electrical and Electronic Engineers., 1988.

[225] S. Narayanan, A. Potamianos, and H. Wang, "Multimodal systems for children: Building a prototype," in *Proc. European Conf. on Speech Communication and Technology*, (Budapest, Hungary), Sept. 1999.

[226] Neal, J. and Shapiro, S., "Intelligent Multi-Media Interface Technology." In J. Sullivan and S, Tyler (Eds.) Intelligent User Interfaces . Addison-Wesley. 11-43, 1991.

[227] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, "Dynamic bayesian networks for audio-visual speech recognition," *EURASIP Journal on Applied Signal Processing*, no. 11, pp. 1–15, 2002.

[228] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou, "Audio-visual speech recognition: Final workshop 2000 report," tech. rep., Center for Language and Speech Processing, The Johns Hopkins University, 2000.

[229] Newman W.M., "A System for Interactive Graphical Programming, in AFIPS Spring Joint Computer Conference. 1968. 28. pp. 47-54."

[230] Nielsen J., "Multimedia and Hypertext: the Internet and Beyond. 1995, Boston: Academic Press Professional. 480."

[231] Nigay, L., Coutaz, J., "A design space for multimodal systems: concurrent processing and data fusion." Human Factors in Computing Systems. In Proceedings of INTERCHI'93, ACM Press: 172-178, 1993.

[232] Nikolov S. G., Bull D. R., Glichrist I. D. (2002), "Gaze-contingent multi-modality displays of multi-layered geographical maps." Proc. of the 5th Intl. Conf. on Numerical Methods and Applications (NM&A02), Symposium on Numerical Methods for Sensor Data Processing, Borovetz, Bulgaria, 2002.

[233] Numajiri T., A. Nakamura, and Y. Kuno (2002), "Speed browser controlled by eye movements." IEEE Int Conf. on Multimedia and Expo, August 26-29, Lausanne, 2002.

[234] Olive, J. P., Greenwood, A., and Coleman, J. (1993)., "Acoustics of american english speech, a dynamic approach. springer-verlag.," 1993.

[235] Olsen Jr. D.R. and Dempsey E.P., "Syngraph: A Graphical User Interface Generator, in Proceedings SIGGRAPH'83: Computer Graphics. 1983. Detroit, MI: 17. pp. 43-50."

[236] Open Software Foundation (1993), "OSF/Motif Guide Style Guide. Englewood Cliffs, NJ:Prentice Hall ."

[237] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[238] Oviatt, S., "Mutual disambiguation of recognition errors in a multimodal architecture." In Proceedings of Conference on Human Factors in Computing Systems: CHI '99, New York, N.Y., ACM Press: 576-583, 1999.

[239] Oviatt, S., "Ten Myths of Multimodal Interaction." Communications of the ACM, 1999.

[240] Oviatt, S., "Multimodal Interfaces." Chapther to appear in Handbook of Human-Computer Interaction, (ed. by J. Jacko & A. Sears), Lawrence Erlbaum: New Jersey, 2002.

[241] Palay, A.J., et al, "The Andrew Toolkit - An Overview, in Proceedings Winter Usenix Technical Conference. 1988. Dallas, Tex: pp. 9-21."

[242] Pallett, D., Fiscus, J., Fisher, W., Garofolo, J., Lund, B., and Prysbocki, M. (1994), "Benchmark tests for the ARPA spoken language program. In Proceedings of the 1994 ARPA Human Language Technology Workshop, pages 49-74, Princeton, New Jersey. Advanced Research Projects Agency, Morgan Kaufmann."

[243] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. Int'l Conf. on Comp. Vision*, pp. 555–562, 1998.

[244] Parasuraman, R., Mouloua, M., and Hilburn, B., "Adaptive aiding and adaptive task allocation enhance human-machine interaction." In M.W. Scerbo and M. Mouloua (Ed.), Proceedings of the Proceedings of the Third Conference on Automation Technology and Human Performance (119-123). Norfolk, VA, 1998.

[245] A. Pargellis, H.-K. Kuo, and C.-H. Lee, "Automatic application generator matches user expectations to system capabilities," in *Proc. ESCA Workshop Interact. Dialog. Multi-Modal Syst.*, (Kloster Irsee, Germany), June 1999.

[246] A. Pargellis, H.-K. Kuo, and C.-H. Lee, "Automatic dialogue generator creates user defined applications," in *Proc. European Conf. on Speech Communication and Technology*, (Budapest, Hungary), Sept. 1999.

[247] A. Pargellis, Q. Zhou, A. Saad, and C.-H. Lee, "A language for creating speech applications," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[248] T. Paymans, J. Lindenberg, and M. Neerincx, "Usability trade-offs for adaptive user interfaces: Ease of use and learnability." Proceedings of the 9th international conference on Intelligent user interface, 2004.

[249] Pazzani, M., Muramatsu, J., and Billsus, D., "Syskill & webert: Identifying interesting web sites." Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 54-61. Portland, OR: AAAI Press, 1996.

[250] Pereira, F. C. N. and Schabes, Y. (1992), "Inside-outside reestimation from partially bracketed corpora.." In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, pages 128-135, University, 1992.

[251] E. Petajan, *Automatic Lipreading to Enhance Speech Recognition.* PhD thesis, Univ. of Illinois, Urbana-Campaign, 1984.

[252] R. Pieraccini, K. Dayanidhi, J. Bloom, J.-G. Dahan, M. Phillips, B. Goodman, and K. Prasad, "Multimodal conversational systems for automobiles," *Communications of the ACM*, vol. 47, pp. 47–49, Jan. 2004.

[253] R. Pieraccini, E. Levin, and W. Eckert, "AMICA: the AT&T mixed initiative conversational architecture," in *Proc. European Conf. on Speech Communication and Technology*, (Rhodos, Greece), Sept. 1997.

[254] Plaisant, Catherine, Rose, Anne, Milash, Brett, Widoff, Seth, and Shneiderman, Ben, "Lifelines: Visualizing personal histories." *Prof. of CHI'96 Conference: Human Factors in Computing systems*, ACM, New York (1996), 221-227, 518.

[255] Pollard, C. and Sag, I. (1994), "Head-driven phrase structure grammar.." Center for the Study of Language and Information (CSLI) Lecture Notes. Stanford University Press and University of Chicago Press., 1994.

[256] Pomplun M. & H. Ritter (1999), "A three-level model of comparative visual search." In M. Hahn & S. C. Stoness, (Eds.), Proceedings of the 21st Annual Conference of the Cognitive Science Society, 543-548, 1999.

[257] A. Potamianos and S. Narayanan, "Spoken dialog systems for children," in *Proc. Internat. Conf. on Acoust., Speech, and Signal Process.*, (Seattle,Washington), pp. 197–201, May 1998.

[258] A. Potamianos, G. Riccardi, and S. Narayanan, "Categorical understanding using statistical n-gram models," in *Proc. European Conf. on Speech Communication and Technology*, (Budapest, Hungary), Sept. 1999.

[259] G. Potamianos, H. Graf, and E. Cosatto, "An image transform approach for HMM based automatic lipreading," in *Proc. Int'l Conf. on Image Proc.*, vol. III, pp. 173–77, 1998.

[260] G. Potamianos and C. Neti, "Automatic speechreading of impaired speech," in *Int'l Conf. on Auditory-Visual Speech Processing*, pp. 177–182, 2001.

[261] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: An overview," in *Issues in Visual and Audio-Visual Speech Processing* (G. Bailly, E. Vatikiotis-Bateson, and P. Perrier, eds.), ch. 10, MIT Press, 2004.

[262] Privitera C.M., Stark L. W. (2000), "Algorithms for defining visual regions of interest: Comparison with eye fixations." IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 22, No 9, pp 970-982, 2000.

[263] Rabiner, L. R., "A tutorial on hidden markov models and selected applications in speech recognition.." Proceedings of the IEEE, 77(2):2570-286., 1989.

[264] Rabiner, L. R. Juang, B. H., "Fundamentals of Speech Recognition.." Prentice Hall, Englewood Cliffs, NJ, 1993.

[265] Rao, Ramana and Card, Stuart, K., "The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information." *Proc CHI'94 Conference: Human Factors in Computing Systems*, ACM, New York (1994), 318-322.

[266] H. R.D., "Supporting Concurrency, Communication and Synchronization in Human- Computer Interaction – The Sassafras UIMS. ACM Transactions on Graphics, 1986. 5(3) pp. 179-210."

[267] Robert W. Scheifler and Jim Gettys, "The X Window System. ACM Transactions on Graphics 5, 2 (April 1986), 79-109 ."

[268] Robertsob George G. and Mackinlay, Jock D., "The document lens." *Proc.1993 ACM User Interface Software and Technology*, ACM New York (1993), 101-108.

[269] Robertson, George G., Card, Stuart., and Mackinlay, Jock D.,, "Information visualization using 3-d interactibe annimation." *Communications of the ACM*, 36, 4, (April 1993), 56-71.

[270] Robinson, D. A. (1963), "A method of measuring eye movement using a scleral search coil in a magnetic field." IEEE Transactions on Biomedical Electronics, BME-10, 137-145, 1963.

[271] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake, "Computationally efficient face detection," in *Proc. Int'l Conf. on Comp. Vision*, vol. II, pp. 695–700, 2001.

[272] Ron ColeJoseph Mariani, Hans Uszkoreit, Hans Uszkoreit, Giovanni Batista Varile, Annie Zaenen, Antonio Zampolli and Victor Zue, "Survey of the State of the Art in Human Language Technology. Cambridge University Press and Giardini 1997."

[273] L. Rothrock, R. Koubek, F. Fuchs, M. Haas, and G. Salvendy, "Review and reappraisal of adaptive interfaces: Towards biologically-inspired paradigms." Theoritical Issues in Ergonomics Science, 2002.

[274] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–38, Jan. 1998.

[275] Sagisaka, Y., Kaiki, N., Iwahashi, N., and Mimura, K. (1992)., "Atr - talk speech synthesis system.." In Proceedings of the 1992 International Conference on Spoken Language Processing, volume 1, pages 483-486, Ban , Alberta, Canada. University of Alberta., 1992.

[276] Salvucci D. D. (2000), "An interactive model-based environment for eye-movement protocol analysis and visualization." Proceedings of the symposium on Eye Tracking Research & Applications, p.57-63, Palm Beach Gardens, Florida, United States, 2000.

[277] Salvucci, D. D., and Goldberg, J. H. (2000), "Identifying fixations and saccades in eye-tracking protocols." In Proceedings of the Eye Tracking Research and Applications Symposium (pp. 71-78). New York: ACM Press, 2000.

[278] R. Sarvas, "Media content metadata and mobile picture sharing," in *Proceedings of the Finnish Artificial Intelligence Conference STeP 2004* (E. Hyvönen, T. Kauppinen, M. Salminen, K. Viljanen, and P. Ala-Siuru, eds.), vol. 2 of *Conference Series – No 20*, pp. 131–146, Finnish Artificial Intelligence Society, Sept. 2004.

[279] Schabes, Y., Roth, M., and Osborne, R. (1993), "Parsing the wall street journal with the inside-outside algorithm.." In Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics, Utrecht University, The Netherlands. European Chapter of the Association for Computational Linguistics., 1993.

[280] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. I, pp. 746–751, 2000.

[281] Schneiderman, B., "Direct manipulation for comprehensible, predictable and controllable user interface." Proceedings of 1997 International conference on Intelligent User Interfaces. ACM Press, 1997.

[282] Schnell, T., Wu, T., (2000), "Applying eye tracking as alternative approach for activation of controls and functions in aircraft." Proceedings of the 5th International Conference On Human Interaction with Complex Systems (HICS), April, 30 - May, 2, 2000, Urbana, Illinois, USA, pp 113, 2000.

[283] Schwartz, R., Chow, Y., and Kubala, F. (1987)., "Rapid speaker adaption using a probabalistic spectral mapping.." In Proceedings of the 1987 International Conference on Acoustics, Speech, and Signal Processing, pages 633-636, Dallas. Institute of Electrical and Electronic Engineers., 1987.

[284] S. Sclaroff and J. Isidoro, "Active blobs," in *Proc. Int'l Conf. on Comp. Vision*, pp. 1146–1153, 1998.

[285] Sears, Andrew, Revis, S. Doreen, Jean, Crittenden, Robert, Shneiderman, and Ben, "Investigating touchscreen typing: The effect of keyboard size on typing speed, *Behaviour & Information Technology*, 12, 1 (Jan-Feb 1993),17-22."

[286] Selker Ted, Lockerd Andrea, Martinez Jorge (2001), "Eye-r, a glasses-mounted eye motion detection interface." CHI '01 extended abstracts on Human factors in computer systems, Seattle, Washington, 2001.

[287] S. Seneff et al, "Galaxy-II: A reference architecture for conversational system development," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[288] Seneff, S., Lau, R., Polifroni, J., "Organization, Communication, And Control In The Galaxy-II Conversational System." In Proceedings of Eurospeech'99, Budapest, Hungary: 1271-1274, 1999.

[289] Shaikh, A., Juth, S., Medl, A., Marsic, I., Kulikowski, C., Flanagan, J., "An architecture for multimodal information fusion." Proceedings of the Workshop on Perceptual User Interfaces (PUI 97), 91-93. Banff, Canada, 1997.

[290] Shardanand, U., and Maes, P., "Social information filtering: Algorithms for automating word of mouth." Proceedings of the Conference on Human Factors in Computing Systems, pp. 210-217. Denver, CO: ACM Press, 1995.

[291] Sherr, "Sol (Editor), *Input devices.*" Academic Press, San Diego, CA, 1988.

[292] Shieber, S. M. (1983), "Sentence disambiguation by a shift-reduce parsing technique.." In Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, pages 113-118, Cambridge, Massachusetts. Association for Computational Linguistics., 1983.

[293] Shneiderman, B., "Designing the user Interface: Strategies for Effective Human-Computer Interaction, *Addison-Wesley, Don Mills Ontario, 1992, pp 522-526, p 524* ."

[294] Shneiderman B., "Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer, 1983. 16(8) pp. 57-69. August 1983."

[295] Shumin Zhai and Paul Milgram, "Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices." In Proceedings of the SIGCHI conference on Human factors in computing systems, Los Angeles, California,Pages: 320-327, 1998.

[296] Siroux, J., Guyomard, M., Multon, F. and Remondeau, C., "Oral and Gestural Activities of the Users in the Georal System." In Proc. of the Intl. Conf. on Cooperative Multimodal Communication, vol. 2, 1995, 287–298, 1995.

[297] R. W. Smith and D. R. Hipp, *Spoken Natural Language Dialog Systems.* New York, NY: Oxford University Press, 1994.

[298] Somberg, B.L., "A comparison of ruled-based and position ally constant arrangements of computer menu items." Proceedings of CHI & GI 87 Conference on Human Factors in Computing systems. New-York: ACM, 1987.

[299] Spence, Robert and Apperley, Mark, "Data base investigation: An office envirenment for the professional." *Behaviour & Information Technology*, 1,1 43-45, 1982.

[300] Stockham, T. G., J., Connon, T. M., and Ingebretsen, R. B., "Blind deconvolution through digital signal processing.." Proceedings of the IEEE, 63(4):678-692., (1975).

[301] D. Stork and M. Hennecke, eds., *Speechreading by Humans and Machines.* Berlin, Germany: Springer, 1996.

[302] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39–51, Jan. 1998.

[303] Sutherland, I. E., "Sketchpad, A Man-Machine Communication System, Lincoin Report TR-396, January, 1963. (Ph.D. Thesis in Department of Electrical Engineering, M. I. T.) also published in condensed form in the Proceedings of the 1963 Spring Joint Computer Conference in Detroit."

[304] Sutherland, I.E., " SketchPad: A Man-Machine Graphical Communication System,in AFIPS Spring Joint Computer Conference. 1963. 23. pp. 329-346 ."

[305] S. Sutton et al, "Universal speech tools: The CLSU toolkit," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[306] N. Suzuki, K. Ishii, and M. Okada, "Organizing self-motivated dialogue with autonomous creatures," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[307] Takeda, K., Abe, K., and Sagisaka, Y. (1992)., "On the basic scheme and algorithms in non-uniform unit speech synthesis.." In Bailly, G. and Benoit, C., editors, Talking Machines: Theories, Models, and Designs, pages 93-105. Elsevier Science., 1992.

[308] Takehiko Ohno, Naoki Mukawa and Atsushi Yoshikawa (2002), "Freegaze: a gaze tracking system for everyday gaze interaction." Eye Tracking Research & Applications symposium, pp.125-132, 2002.

[309] Thomas J.J.,, " Graphical Input Interaction Technique (GIIT) Workshop Summary. SIGGRAPH Computer Graphics, 1983. 17(1) pp. 5-30. (Summary of first ZBattelle Seattle UIMS Workshop June 2-4, 1982) ."

[310] Trumbly, J.E., Arnett, K.P., and Johnson, P.C., "Productivity gains via an adaptive user interface." Journal of human-computer studies, 40 (1994), pp. 63-81, 1994.

[311] Turunen, M., *Jaspis - A Spoken Dialogue Architecture and its Applications.* PhD thesis, University of Tampere, Department of Information Studies, 2004.

[312] Turunen, M., Hakulinen, J., "Jaspis2 - An Architecture For Supporting Distributed Spoken Dialogues." In Proceedings of Eurospeech 2003: 1913-1916, 2003.

[313] Tweedie, Lisa, Spence, Robert, Dawkes, Huw, and Su, Hua, "Externalizing abstract mathematical models." *Proc. of CHI'96 Conference: Human Factors in Computing Systems,*ACM, New York (1996), 406-412.

[314] Unisys Corporation, *Unisys Natural Language Speech Assistant.* 1999.

[315] Van Mulken S., Andre E., and Mtiller J., "The Personal Effect: How Substantial is it? *in Proc. of HCI98, Sheffield, England, 53-66, 1998.*"

[316] Vander Zanden, B.T., " Constraint Grammars–A New Model for Specifying Graphical Applications, in Proceedings SIGCHI'89: Human Factors in Computing Systems. 1989. Austin, TX: pp. 325-330."

[317] J. Vermaak, A. Blake, M. Gangnet, and P. Perez, "Sequential monte carlo fusion of sound and vision for speaker tracking," in *Proc. Int'l Conf. on Comp. Vision*, pp. 741–746, 2001.

[318] Verplank B. (1988), "Designing Graphical User Interfaces. Proceedings:CHI'88.May 15."

[319] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. I, pp. 511–518, 2001.

[320] Virvou, M., "Automatic reasoning and help about human errors in using an operating system." Interacting with Computers, 11, 545-573, 1999.

[321] Wahlster, W., Reithinger N., Blocher, A., "SmartKom: Multimodal Communication with a Life-Like Character." In Proceedings of Eurospeech 2001: 1547-1550, 2001.

[322] M. Walker, D. Litman, C. Kamm, and A. Abella, "Evaluating spoken dialogue agents with paradise: Two case studies," *Computer Speech and Language*, pp. 317–347, 1998.

[323] K. Wang, "An event driven model for dialogue systems," in *Internat. Conf. Speech Language Processing*, (Sydney, Australia), Dec. 1998.

[324] Ward D.J. and MacKay D.J.C. (2002), "Fast hands-free writing by gaze direction." Nature 418 pp 838, 2002.

[325] Wasserman A.I. and Shewmake, D.T., "Rapid Prototyping of Interactive Information Systems. ACM Software Engineering Notes, 1982. 7(5) pp. 171-180 ."

[326] Weiser, M., "Some Computer Science Issues in Ubiquitous Computing. CACM,1993. 36(7) pp. 74-83. (July 1993)."

[327] A. Wilhelm, Y. Takhteyev, R. Sarvas, N. V. House, and M. Davis, "Photo annotation on a camera phone," in *Extended abstracts of the 2004 conference on Human factors and computing systems*, pp. 1403–1406, ACM Press, 2004.

[328] William K. English, Douglas C. Engelbart, and Melvyn L. Berman, "Display-Selection Techniques for Text Manipulation, *IEEE Transactions on Human Factors in Electronics*, March 1967, Vol. HFE-8, No. 1, pp. 5-15 ."

[329] Wise, James A., Thomas, James, J., Pennock, Kelly, Lantrip, David, Pottier, Marc, Schur, Anne and Crow, Vern, "Visualizing the non-visual: Spatial analysis and interaction with information from text documents." *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 51-58.

[330] Xin Fan, Xing Xie, Wei-Ying Ma, Hong-Jiang Zhang, He-Qin Zhou (2003), "Visual attention based image browsing on mobile devices." IEEE International Conference on Multimedia and Expo., Vol.I, pp 53-56, Baltimore, MD, USA, 2003.

[331] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, Jan. 2002.

[332] S. Young, "A review of large-vocabulary continuous-speech recognition," *IEEE Signal Processing Magazine*, pp. 45–57, Sept. 1996.

[333] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK book (for HTK version 3.2)," tech. rep., Cambridge University Engineering Department, Dec. 2002.

[334] A. Yuille, P. Hallinan, and D. Cohen, "Feature extraction from faces using deformable templates," *Int'l Journal of Comp. Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[335] S. Zhai, "Performance Optimization of Virtual Keyboards *HUMAN-COMPUTER INTERACTION, 2002, Volume 17, Lawrence Erlbaum Associates.*"

[336] S. Zhai, M. A. Hunter, and B. A. Smith, "The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design," in *UIST*, pp. 119–128, 2000.

[337] S. Zhai, A. Sue, and J. Accot, "Movement model, hits distribution and learning in virtual keyboarding," in *Proceedings of ACM CHI'2002 Conference on Human Factors in Computing Systems*, pp. 17–24, 2002.

[338] Q. Zhou, C.-H. Lee, W. Chou, and A. Pargellis, "Speech technology integration and research platform: A system study," in *Proc. European Conf. on Speech Communication and Technology*, (Rhodes, Greece), Sept. 1997.

[339] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Joint audio-visual tracking using particle filters," *EURASIP J. Applied Signal Processing*, vol. 11, pp. 1154–1164, 2002.