TeleMorph & TeleTuras: Bandwidth determined Mobile MultiModal Presentation

Anthony Solon

Supervisors: Prof. Paul Mc Kevitt, Kevin Curran

First Year Report

Date: December, 2003

Presented as a requirement for Ph.D. in
School of Computing and Intelligent Systems
Faculty of Engineering
University of Ulster, Magee
Email: aj.solon@ulster.ac.uk

Abstract

The objective of the work described here is the development of a mobile intelligent multimedia presentation architecture called TeleMorph and a respective application system called TeleTuras. TeleMorph will be able to dynamically generate a multimedia presentation from semantic representations using output modalities (speech, text, graphics) that are determined by constraints that exist on a mobile device's wireless connection, the mobile device itself and also those limitations experienced by the end user of the device. TeleTuras, a tourist information application for the city of Derry, will be developed as a testbed for TeleMorph.

Previous research in the related areas of mobile intelligent multimedia systems, intelligent multimedia presentation systems, wireless telecommunications, network-adaptive multimedia models, semantic representation, fusion and coordination of modalities, Cognitive Load Theory (CLT), and Causal Probabilistic Networks (CPNs) are surveyed, and corresponding computer systems are explored. Most of the current work on mobile intelligent multimedia presentation systems determines the output presentation modalities by only considering constraints imposed by the mobile client device or the end user of the system. These systems fail to take the state of the wireless network into account when choosing output presentation modalities. TeleMorph will improve upon existing systems by dynamically morphing the output presentation modalities depending on: (1) available network bandwidth (also network latency, packet loss and error rate); (2) available client device resources (mobile device display resolution, available memory and processing power, available output abilities); (3) end user modality preferences and associated costs incurred in downloading a presentation; (4) the cognitive load of the end user and whether the function of the presentation is directed towards end-user retention (e.g. a city tour) or is intended solely for informative purposes (e.g. about an interesting sight nearby). Causal Probabilistic Networks are an approach to reasoning and decision making that will be utilised in TeleMorph's architecture. The aforementioned constraints contribute to the automatic adaptation features of TeleMorph, which will consider a union of their effects using Causal Probabilistic Networks. Hence the main area of contribution in TeleMorph is its awareness of available bandwidth and the union of this with other relevant constraints. TeleTuras, the testbed application for TeleMorph will communicate TeleMorph-adapted presentations to tourists, focusing on the output modalities used to communicate information and also the effectiveness of this communication. Applying a collection of common questions will test TeleTuras. These questions will be accumulated by asking prospective users/tourists what they would require from a tourist information aid such as TeleTuras.

TeleMorph will be developed as a client-server architecture using existing software tools such as Java 2 Micro Edition (J2ME), Synchronised Multimedia Integration Language (SMIL) editors/players, HUGIN and JATLite. The J2ME programming environment provides a set of tools and Application Programming Interfaces (APIs) to develop speech (using the Java Speech API) graphics (using graphics APIs) and text (using high level graphics API) input/output modules for the client application. The graphics API in J2ME will also allow for haptic input by the user. Other J2ME tools that will be reused in the implementation of the client include the sysInfo application (for detecting device status) and wireless networking APIs (for communicating input to the server). On the client side of TeleMorph an appropriate intelligent multimedia agent (e.g. MS Agent) will be used to implement an actor for more effective communication. The SMIL language (XML based) will be used as the semantic representation language in TeleMorph whilst possibly reusing elements of the MPEG-7 standard (also XML based). To present SMIL based multimedia presentations on the mobile client device (e.g. Compaq iPaq) a SMIL compatible media player will be used. TeleMorph's server-side presentation design module will use the HUGIN development environment which is based on the Causal Probabilistic Network (CPN) approach to reasoning and decision making. HUGIN will analyse a union of all relevant constraints imposed on TeleMorph and decide on the optimal multimodal output presentation. Middleware such as JATLite or Open Agent Architecture will provide a solution for integration and interoperation between TeleMorph's various modules.

TeleTuras will enable input queries in a variety of modalities whether they are combined or used individually. Queries can be directly related to the user's position and movement direction enabling questions/commands such as: "Where is the Millenium forum?" "Take me to the GuildHall". TeleTuras will be capable of communicating TeleMorph-adapted presentations in response to queries consisting of: route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. Hence, TeleTuras will provide tourists with instantaneous access to relevant information.

Keywords: TeleMorph, TeleTuras, mobile intelligent multimedia, intelligent multimedia generation, intelligent multimedia presentation, multimodal output, causal probabilistic networks, mobile network constraints, mobile-device awareness, bandwidth awareness, artificial intelligence

Acknowledgements

I would like to express my gratitude to Prof. Paul Mc Kevitt and Kevin Curran for their advice and guidance throughout the first year of my Ph.D. Also I am grateful to the other members of the Intelligent Multimedia group here at Magee, Minhua Eunice Ma and Dimitrios Konstantinou, from our discussions I have benefited from their knowledge of IntelliMedia. Finally I want to thank my family and close friends for their continuous encouragement and support.

Table of Contents

1. Introduction

1.1 Background

Whereas traditional interfaces support sequential and un-ambiguous input from keyboards and conventional pointing devices (e.g., mouse, trackpad), intelligent multimodal interfaces relax these constraints and typically incorporate a broader range of input devices (e.g., spoken language, eye and head tracking, three dimensional (3D) gesture) (Maybury, 1999). The integration of multiple modes of input as outlined by Maybury allows users to benefit from the optimal way in which human communication works. "Put-That-There" (Bolt, 1987) was one of the first intelligent multimodal interfaces. The interface consisted of a large room, one wall of which was a back projection panel. Users sat in the center of the room in a chair wearing magnetic position sensing devices on their wrists to measure hand position. Users could use speech, and gesture, or a combination of the two to add, delete and move graphical objects shown on the wall projection panel. Mc Kevitt (1995a,b, 1996a,b) focuses on the problem of integrating natural language and vision processing, whilst Mc Kevitt et al. (2002) concentrates on language, vision and music, identifying cognitive patterns that underlie our competence in these disparate modes of thought. Maybury and Wahlster (1998) focus on intelligent user interfaces and according to them the generation process of an intelligent multimedia presentation system can be divided into several co-constraining processes including: the determination of communicative intent, content selection, structuring and ordering, allocation to particular media, media realisation, coordination across media, and layout design.

Whereas humans have a natural facility for managing and exploiting multiple input and output media, computers do not. To incorporate multimodality in user interfaces enables computer behaviour to become analogous to human communication paradigms, and therefore the interfaces are easier to learn and use. Since there are large individual differences in ability and user preferences for different modes of communication, a multimodal interface permits the user to exercise selection and control over how they interact with the computer (Fell et al., 1994; Karshmer & Blattner, 1998). In this respect, multimodal interfaces have the potential to accommodate a broader range of users than traditional graphical user interfaces (GUIs) and unimodal interfaces- including users of different ages, skill levels, native language status, cognitive styles, sensory impairments, and other temporary or permanent handicaps or illnesses. Interfaces involving spoken or pen-based input, as well as the combination of both, are particularly effective for supporting mobile tasks, such as communications and personal navigation. Unlike the keyboard and mouse, both speech and pen are compact and portable. When combined, people can shift these input modes from moment to moment as environmental conditions change (Holzman 1999).

Implementing multimodal user interfaces on mobile devices is not as clear-cut as doing so on ordinary desktop devices. This is due to the fact that mobile devices are limited in many respects: memory, processing power, input modes, battery power, and an unreliable wireless connection. This project will research and implement a framework, TeleMorph, for multimodal interaction in mobile environments taking these and other issues into consideration, but primarily– fluctuating bandwidth. TeleMorph output will be bandwidth dependent, with the result that output from semantic representations is dynamically morphed between modalities or combinations of modalities depending on available bandwidth. Further to the consideration of limited bandwidth, the following constraints will contribute to the automatic adaptation features of TeleMorph:

- mobile device constraints
    - mobile device's display resolution
    - memory availability
    - processing power
    - output abilities
- end user constraints
    - domain/environment
    - mobility (stationary/high-speed)
    - end-user's cognitive load
    - the presentation's purpose (educational/ informative)

According to Pedersen & Larsen (2003), automatic modality shifting by a system can lead to user annoyance so it seems necessary to integrate the option for user-controlled modality shifting in TeleMorph to avoid this problem. This will be achieved by also enabling the end user to choose their preferred output modalities leading to greater user-acceptance of the presentation (Elting et al. 2001). Another reason for enabling user-control is that next generation mobile network customers will be charged by the amount of data (Mega Bytes) that they receive to their mobile device (although some carriers intend to charge per-event (e.g. per- email download) these events are graded on size i.e. Mega Bytes). TeleMorph will permit the user to select various output modality combinations and to manually adjust the quality of each modality, as well as providing the user with immediate feedback on the affect to the total cost incurred, thus enabling the user to manage the presentation more effectively. The cognitive load of the end user will be considered by TeleMorph with focus on whether the function of the output presentation is directed towards end-user retention (e.g. a city tour), when the most effective presentation modalities should be used, or is intended solely for informative purposes (e.g. an interesting sight nearby), when the most appealing presentation modalities should be used. This approach is adapted from Baddeley's Cognitive Load Theory (Baddeley & Logie 1999, Sweller et al. 1998). TeleMorph will provide output that adheres to good usability practice, resulting in suitable throughput of information and context sensitive modality combinations in keeping with Cognitive Load Theory. Causal Probabilistic Networks (CPNs) (Jensen & Jianming 1995) are an example of an approach to reasoning and decision making and TeleMorph's presentation design module will utilise this technique. Using this technique a union of the aforementioned constraints imposed on TeleMorph will be analysed and the optimal multimodal output presentation will be determined.

To demonstrate the effectiveness of this research a tourist information aid called TeleTuras is proposed that will provide a testbed for TeleMorph. TeleTuras will communicate TeleMorph-adapted presentations to tourists, consisting of: route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. The main focus in TeleTuras will be on the output modalities used to communicate this information and also the effectiveness of this communication.

1.2 Objectives of this research

To develop a mobile intelligent multimedia architecture, TeleMorph, that dynamically morphs between output modalities depending primarily on available network bandwidth. The aims of this research are to:

- Determine a mobile device's output presentation (unimodal/multimodal) depending on the network bandwidth available to the mobile device connected to the system. Other secondary issues that will be considered in TeleMorph include: network latency and bit error rate, mobile device constraints (display, available output modalities, memory, processing power) and user constraints (modality preferences, cost incurred, user's cognitive load determined by Cognitive Load Theory), which will be combined and utilised to determine morphing behaviour between output modalities during presentation design.
- Use Causal Probabilistic Networks (CPNs) to analyse a union of all relevant constraints imposed on TeleMorph and decide on the optimal multimodal output presentation.
- Implement TeleTuras, a tourist information guide for the city of Derry and integrate the solution provided by TeleMorph, thus demonstrating its effectiveness.

TeleTuras will communicate TeleMorph-adapted presentations to tourists consisting of: route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. The main focus in TeleTuras will be on the output modalities used to communicate this information and also the effectiveness of this communication. The aims entail the following objectives:

- Receive and interpret questions from the user using Media Analysis module.
- Map questions to multimodal semantic representation.
- Match multimodal representation to knowledge base to retrieve answer.
- Map answers to multimodal semantic representation.
- Monitor user preference or client side choice variations.
- Query bandwidth status.
- Detect client device constraints and limitations.
- Combine affect of all constraints imposed on system using CPNs.
- Generate optimal multimodal presentation based on bandwidth constraint data (whilst also considering the union of all relevant constraints).

TeleTuras will be tested with a corpus of common questions. These questions will be accumulated by asking prospective users/tourists what they would require from a tourist information aid like TeleTuras.

1.3 Area of contribution

TeleMorph will intelligently morph multimodal output presentations depending on available network bandwidth. TeleMorph will be an improvement on previous systems in that it is bandwidth-aware, and also in that it considers bandwidth in a union with other relevant constraints using Causal Probabilistic Networks (CPNs). Thus, TeleMorph's unique contribution has been identified – *Bandwidth determined Mobile Multimodal Presentation*.

In Chapter 2 related research in the areas of wireless telecommunications, mobile intelligent multimedia systems, network-adaptive multimedia models, semantic representation, fusion and coordination of modalities, intelligent multimedia presentation systems, intelligent multimedia interfaces, intelligent multimedia agents, Cognitive Load Theory (CLT) and Causal Probabilistic Networks (CPNs) are investigated. Following this, in Chapter 3, we turn to the system and unit design of the TeleMorph architecture and compare it with related work. Next in Chapter 4 we explore software for implementing client and server modules in TeleMorph and TeleTuras including Java 2 Micro Edition, Synchronised Multimedia Integration Language editors/players, HUGIN, and various middlewares. Finally Chapter 5 concludes this report.

2. Literature review

In the following sections a variety of areas relating to this project proposal will be reviewed. Initially, wireless telecommunications are discussed with brief descriptions of mobile technologies involved in mobile intelligent multimedia systems. Examples of mobile intelligent multimedia systems are then detailed. Following this, network-adaptive multimedia models are synopsised, including sections on transcoding proxies and end-to-end networking approaches. Core issues within intelligent multimedia presentation are then discussed: semantic representation; fusion, integration and coordination of modalities. Intelligent multimedia presentation systems are then discussed. Intelligent multimedia interfaces and intelligent multimedia agents follow, including descriptions of projects researching these areas. This chapter is concluded with a review of Cognitive Load Theory (CLT) and Causal Probabilistic Networks (CPNs).

2.1 Wireless telecommunications

However, intelligent multimedia mobile telecommunication systems are far from realisation taking into consideration the current state of the technology available for accessing mobile networks. Despite this, using Moore's law one could assume that eventually network and device capabilities will be sufficient to support intelligent mobile multimedia applications. Projects focussing on intelligent multimedia applications on mobile devices will be discussed in the following sections, but first some technologies that are necessary to enable mobile information systems similar to TeleTuras are detailed including wireless networks and positioning systems.

Mobile phone technologies have evolved in several major phases denoted by "Generations" or "G" for short. Three generations of mobile phones have evolved so far, each successive generation more reliable and flexible than the previous.

- "1G" wireless technology (Tanaka 2001) was developed during the 1980s and early 1990s. It only provided an Analog voice service with no data services available.
- "2G" wireless technology (Tanaka 2001) uses circuit-based, digital networks. Since 2G networks are digital they are capable of carrying data transmissions, with an average speed of around 9.6K bps (bits per second).
- "2.5G" wireless technology (Tanaka 2001) represents various technology upgrades to the existing 2G mobile networks. Upgrades to increase the number of consumers the network can service while boosting data rates to around 56K bps. 2.5G upgrade technologies are designed to be overlaid on top of 2G networks with minimal additional infrastructure. Examples of these technologies include: General Packet Radio Service (GPRS) (Tanaka 2001) and Enhanced Data rates for Global Evolution (EDGE). They are packet based and allow for "always on" connectivity.

- "3G" wireless technology (Tanaka 2001) will be digital mobile multimedia offering broadband mobile communications with voice, video, graphics, audio and other forms of information. 3G builds upon the knowledge and experience derived from the preceding generations of mobile communication, namely 2G and 2.5G. Although, 3G networks use different transmission frequencies from these previous generations and therefore require a different infrastructure. 3G networks will improve data transmission speed up to 144K bps in a high speed moving environment, 384K bps in a low-speed moving environment, and 2M bps in a stationary environment. Currently the available speeds on 3G networks lies at 100-300Kbps, but it is anticipated it will increase to 1- 4Mbps over time with network and device upgrades.
- "4G" IP based mobile/wireless networks (Constantiou et al. 2001) will incorporate Wireless Personal Area Networks (PANs) and sensor networks to fulfil the requirement of 'anywhere and anytime' ubiquitous services. Speeds are anticipated to reach up to 100M bps.

As Figure 2.1 below summarises, bandwidth can vary from a connectionless device which is experiencing loss of network coverage to 2Mbps for stationary devices connected to a 3G mobile network. The forth generation of mobile networks isn't expected to be realised for some time yet.

Mobile Network	**Bandwidth (bits/sec)**
2G	9.6 K
2.5G (GPRS/ EDGE)	56 K
3G (UMTS)	144 K – 384 K – 2M
4G	100M

Table 2.1: Mobile network generations

There are a number of different positioning systems that can provide varying degrees of precision in positioning. The main systems used today are GPS (Global Positioning System), DGPS (Differential GPS), GLONASS (GLObal Navigation Satellite System) and GSM (Global System for Mobile communications) (see Koch 2000) positioning.

- GPS is a satellite based navigation system built and run by the American Department of Defense (DoD). GPS consists of at least 24 satellites orbiting the earth. The satellites transmit signals that a handheld GPS receiver can use to calculate it's current position. For anti-terrorism reasons a distortion system called Selective Availability (SA) is applied to the GPS signal transmitted by the satellites which alters the GPS positioning capabilities to an average of 20-40 meters.

- DGPS is one way around SA, It consists of placing a GPS receiver on a known location to find the difference between the distorted and actual position measurements.
- GLONASS is the Russian version of GPS but does not use a SA distortion system.
- GSM positioning works by triangulating the signals from cellular phone antennas in a central computer, and thereby estimating the position of the user. One example of this method of position discovery is Enhanced Observed Time Difference (E-OTD).

More detail on these systems (GPS, DGPS, GLONNASS, GSM) can be found in Koch (2000).

2.2 Mobile intelligent multimedia systems

With the advent of 3G(Third Generation) wireless networks and the subsequent increased speed in data transfer available, the possibilities for applications and services that will link people throughout the world who are connected to the network will be unprecedented. One may even anticipate a time when the applications available on wireless devices will replace the original versions implemented on ordinary desktop computers. Malaka (2000, p. 5) states that "the main challenge for the success of mobile systems is the design of smart user interfaces and software that allows ubiquitous and easy access to personal information and that is flexible enough to handle changes in user context and availability of resources." A number of issues need to be addressed before the aforementioned aim can be met:

- Location awareness
- Context awareness
- Interaction metaphors and interaction devices for mobile systems
- Smart user interfaces for mobile systems
- Situation adapted user interfaces
- Adaptation to limited resources
- Fault tolerance
- Service discovery, service description languages and standards

Some projects have already investigated mobile intelligent multimedia systems, using tourism in particular as an application domain. Koch (2000) describes one such project which analysed and designed a position-aware speech-enabled hand-held tourist information system. The system is position and direction aware and uses these facilities to guide a tourist on a sight-seeing tour. Rist (2001) describes a system which applies intelligent multimedia to mobile devices. In this system a car driver can take advantage of online and offline information and entertainment services while driving. The driver can control phone and Internet access, radio, music repositories (DVD, CD-ROMs), navigation aids using GPS and car reports/warning systems. Pieraccini (2002) outlines one of the main challenges of these mobile multimodal user interfaces, that being the necessity to adapt to different situations ("situationalisation"). Situationalisation as referred to by Pieraccini identifies that at different moments the user may be subject to different constraints on the visual and aural channels (e.g. walking whilst carrying things, driving a car, being in a noisy environment, wanting privacy etc.).

EMBASSI (Hildebrand 2000) explores new approaches for human-machine communication with specific reference to consumer electronic devices at home (TVs, VCRs, etc.), in cars (radio, CD player, navigation system, etc.) and in public areas (ATMs, ticket vending machines, etc.). Since it is much easier to convey complex information via natural language than by pushing buttons or selecting menus, the EMBASSI project focuses on the integration of multiple modalities like speech, haptic deixis (pointing gestures), and GUI input and output. Because EMBASSI's output is destined for a wide range of devices, the system considers the effects of portraying the same information on these different devices by utilising Cognitive Load Theory (CLT) (Baddeley & Logie 1999, Sweller et al. 1998). Fink and Kobsa (2002) discuss a system for personalising city tours with user modeling. They describe a user modeling server that offers services to personalised systems with regard to the analysis of user actions, the representation of the assumptions about the user, and the inference of additional assumptions based on domain knowledge and characteristics of similar users. Nemirovsky and Davenport (2002) describe a wearable system called *GuideShoes* which uses aesthetic forms of expression for direct information delivery. GuideShoes utilises music as an information medium and musical patterns as a means for navigation in an open space, such as a street. Cohen-Rose & Christiansen (2002) discuss a system called *The Guide* which answers natural language queries about places to eat and drink with relevant stories generated by storytelling agents from a knowledge base containing previously written reviews of places and the food and drink they serve. Oviatt et al. (2000) explain QuickSet a wireless, handheld, collaborative multimodal system that enables a user to formulate a military scenario by creating, positioning and editing units on a map with speech, pen-based gestures and direct manipulation. Descriptions of mobile intelligent multimedia systems will be given in the following sections. SmartKom, DEEP MAP, CRUMPET, VoiceLog and MUST will be described, giving details of the design and example interactions for each system.

2.2.1 SmartKom

SmartKom (Wahlster 2001, 2003, Wahlster et al. 2001a,b, SmartKom Consortium 2003, Reithinger 2001) is a multimodal dialogue system currently being developed by a consortium of several academic and industrial partners. The system combines speech, gesture and facial expressions on the input and output side. The main scientific goal of SmartKom is to design new computational methods for the integration and mutual disambiguation of different modalities on a semantic and pragmatic level. SmartKom is a prototype system for a flexible multimodal human-machine interaction in two substantially different mobile environments, namely pedestrian and car. The system enables integrated trip planning using multimodal input and output. An example spoken navigation dialogue with SmartKom-mobile follows (U – user, SK – SmartKom response):

SK: “Please select a parking place from the map.”
U: “SmartKom, Please look for the nearest parking lot.”
SK: “The parking garage at the main station provides 300 slots. Opening hours are from 6am to 11pm. Do you want to get there?”
U: “No, please tell me about the next parking option.”
SK: “The market parking lot provides 150 slots. It is open 24 hours a day. Do you want to get there?”
U: “Yes, please.”
SK: “I’ll bring you to the market parking lot.”

This example interaction is portrayed in the SmartKom-Mobile interface as shown in Figure 2.1. The presentation includes a map and an autonomous agent that also processes speech.

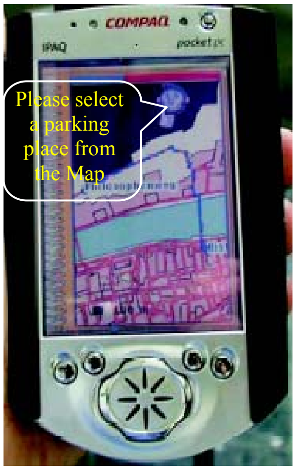


Figure 2.1: Example interaction with the SmartKom-Mobile interface

In a tourist navigation situation a user of SmartKom could ask a question about their friends who are using the same system. E.g. “Where are Tom and Lisa?”, “What are they looking at?” SmartKom is developing an XML-based mark-up language called M3L (MultiModal Markup Language) for the semantic representation of all of the information that flows between the various processing components. The key idea behind SmartKom is to develop a kernel system which can be used within several application scenarios. There exist three versions of SmartKom, which are different in their appearance, but share a lot of basic processing techniques and also standard applications like communication (via email and telephone) and personal assistance (address-book, agenda):

(1) *SmartKom-Mobile* uses a Personal Digital Assistant (PDA) as a front end. Currently, the Compaq iPAQ Pocket PC with a dual slot PC card expansion pack is used as a hardware platform. SmartKom-Mobile provides personalised mobile services like route planning and interactive navigation through a city.
(2) *SmartKom-Public* is a multimodal communication kiosk for airports, train stations, or other public places where people may seek information on facilities such as hotels,

restaurants, and theatres. Users can also access their personalised standard applications via wideband channels.
(3) *SmartKom-Home/Office* realises a multimodal portal to information services. It provides electronic programming guide (EPG) for TV, controls consumer electronic devices like VCRs and DVD players, and accesses standard applications like phone and email.

It is characteristic of the implementation of SmartKom that, in contrast to the research and development within the field of multimedia, it is not the hardware for input and output devices or memory- or transmission-media which have been placed in the foreground; rather the rarely treated aspects of multiple codes and multiple modalities in the context of dialogic interaction are clearly at the center of attention. As the general architecture in Figure 2.2 indicates, SmartKom has been developed as a knowledge-based interface which is based on explicit user models, the discourse structure, the domains and the therein determined functions as well as the available media. The four central components of SmartKom are the modality specific analysers, the multi-modal interaction components, the application interface with its explicit application model as well as the multi-modal media design for the planning of output.

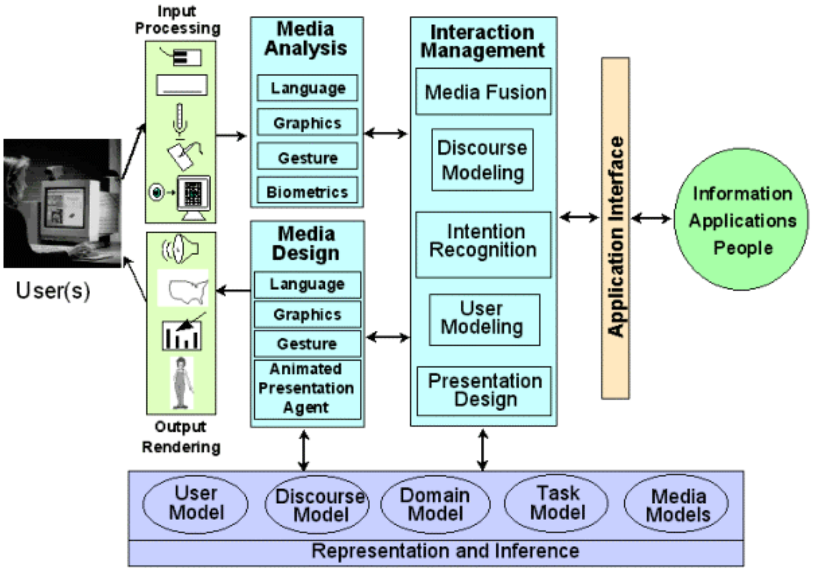


Figure 2.2: SmartKom's architecture

SmartKom is similar to TeleMorph and TeleTuras in that it strives to provide a multimodal information service to the end-user. SmartKom-Mobile is specifically related to TeleTuras in the way it provides location sensitive information of interest to the user of a thin-client device about services or facilities in their vicinity.

2.2.2 DEEP MAP

DEEP MAP (Malaka 2000, 2001, Malaka & Zipf 2000, Malaka et al. 2000, EML 2002) is a prototype of a digital personal mobile tourist guide which integrates research from various areas of computer science: geo-information systems, data bases, natural language processing, intelligent user interfaces, knowledge representation, and more. The goal of Deep Map is to develop information technologies that can handle huge heterogeneous data collections, complex functionality and a variety of technologies, but are still accessible for untrained users. DEEP MAP is an intelligent information system that may assist the user in different situations and locations providing answers to queries such as-Where am I? How do I get from A to B? What attractions are near by? Where can I find a hotel/restaurant? How do I get to the nearest Italian restaurant? It has been developed with two interfaces:

- A web-based interface that can be accessed at home, work or any other networked PC.
- A mobile system that can be used everywhere else.

Both systems, however, are built on identical architectures and communication protocols ensuring seamless information exchanges and hand-overs between the static and mobile system. The main difference between the systems concern the interface paradigms employed and network-related and performance-related aspects. The current prototype is based on a wearable computer called the Xybernaut MA IV. Figure 2.3 below portrays the architecture of DEEP MAP, which consists of agents that reside on three logical layers and are linked to each other through an agent platform.

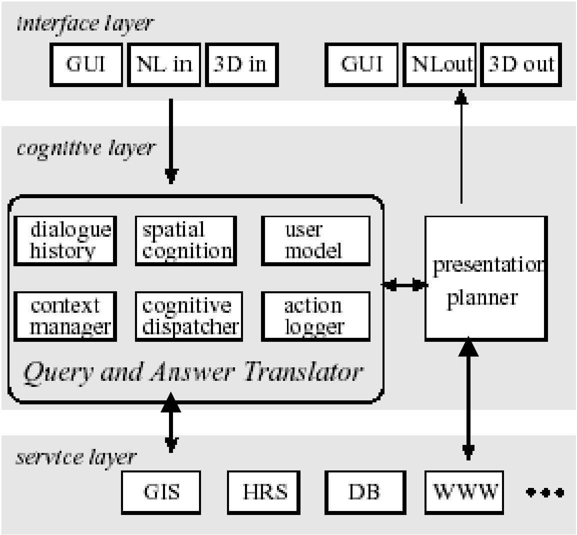
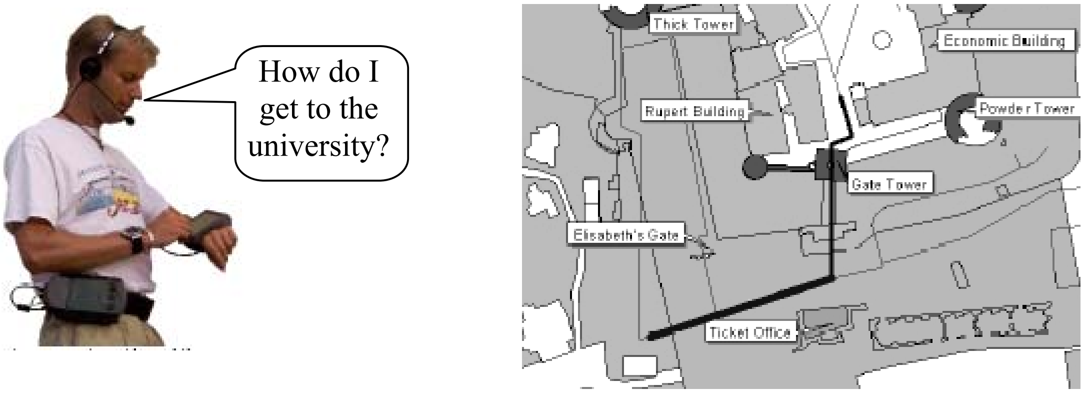


Figure 2.3: DEEP MAP's architecture

The interface layer contains components that directly interact with the user such as the graphical (GUI) or natural language (NL) interface. The cognitive layer components try to understand what the user meant and react accordingly (e.g. the presentation planner), whilst the service layer components provide basic services such as databases (DB), geographic information systems (GIS) and hotel reservation systems (HRS). Examples of input and output in DEEP MAP are given in Figures 2.4a,b respectively.



(a) Example speech input (b) Example map output

Figure 2.4: Example input and output in DEEP MAP

Figure 2.4a shows a user requesting directions to a university within their town using speech input. Figure 2.4b shows an example response to a navigation query, DEEP MAP displays a map which includes the user's current location and their destination, which are connected graphically by a line which follows the roads/streets interconnecting the two. Also, places of interest along the route are displayed on the map.

DEEP MAP is being adapted so that it integrates resource awareness into its architecture. This will be achieved using the RAJA framework. RAJA (Ding et al. 2001) is a Resource-Adaptive Java Agent Infrastructure for the development of resource-adaptive multi-agent systems. It allows for structured programming by using a multi-level architecture to clearly separate domain-specific functionality from resource and adaptation concerns. RAJA is generic since it applies to a wide range of adaptation strategies (e.g. for multi-fidelity systems). RAJA is implemented on top of the FIPA (Foundation for Intelligent Physical Agents) (FIPA 2003) implementations, the FIPA-OS (FIPA-OS 2003) and the JADE (Bellifemine et al. 1999) framework. RAJA supports resource-awareness by offering meta agents that provide infrastructure services addressing resource management. A Resource Monitor has been implemented to keep track of the availability of resources. Currently, methods are available for querying the resource situation of :

- CPU (e.g. the CPU system load, user load, and the idle load),
- memory (e.g. the total amount of (free) memory),
- network bandwidth (basis agents specify socket connections to arbitrary hosts, whose bandwidth should be dynamically monitored).

In order to deliver accurate CPU- and memory-related information, the Resource Monitor makes system calls to the underlying operating system which are platform-dependent. Portability across different platforms is provided by using the Java Native Interface (JNI), while the native libraries for the popular operating systems such as Unix and Windows are written in C. For the measurement of bandwidth, a status object monitoring the amount of data received and dynamically calculating the bandwidth is attached to the input and output stream of each monitored socket. The Resource Monitor of RAJA is also being enhanced to provide statistical information about resource usage (e.g. the average CPU load in the last period of five minutes).

2.2.3 CRUMPET

CRUMPET (Creation of User-friendly Mobile services Personalised for Tourism) (EML 2002, Crumpet 2002, Zipf & Malaka 2001) implements, validates, and tests tourism-related value-added services for nomadic users across mobile and fixed networks. In particular the use of agent technology is evaluated in CRUMPET (in terms of user-acceptability, performance and best-practice) as a suitable approach for fast creation of robust, scalable, seamlessly accessible nomadic services. Figure 2.5 below shows the architecture of CRUMPET. Figure 2.5 shows CRUMPET's Multi-Agent System (MAS) which consists of network, support services and services & users layers. The output of CRUMPET includes an adaptive map system. Map adaptation examples include- culture-specific map colouring, map generalisation, user-orientation dependent maps, focus maps and personalisation. Examples from the CRUMPET map production module are displayed in Figure 2.6.

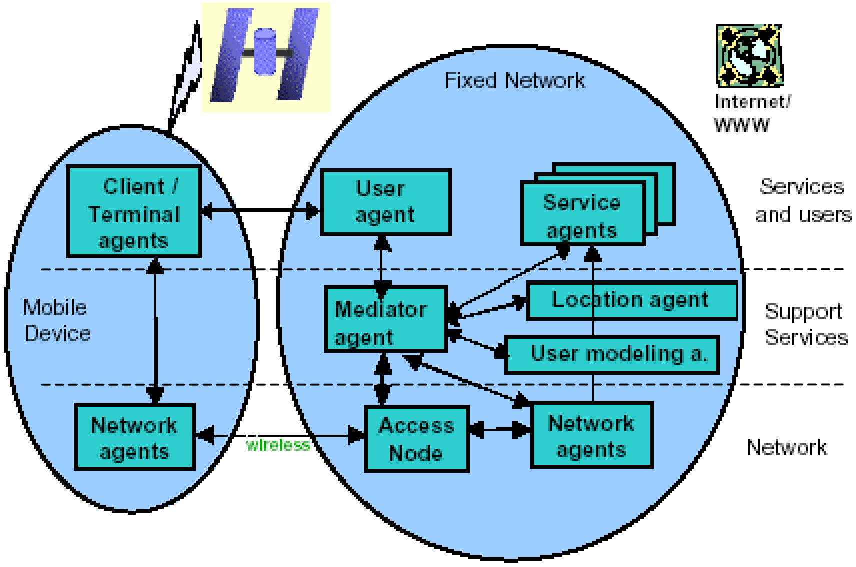


Figure 2.5: CRUMPET's architecture

The implementation is based on a standards-compliant open source agent framework, extended to support nomadic applications, devices, and networks. Main features of the CRUMPET approach include:

- Services that will be trialled and evaluated by multiple mobile service providers.

- Service content that will be tourism-related, supporting intelligent, anytime, anyplace communication suitable for networks like those a typical tourist user might be exposed to now and in the near future.
- Adaptive nomadic services responding to underlying dynamic characteristics, such as network Quality of Service and physical location.
- A service architecture implementation that will be standards-based and made available at the end of the project as (mostly) publicly available open source code.
- Suitability for networks that will be those that a typical tourist user might be exposed to now and in the near future (including IP networks, Wireless LAN, and mobile networks supporting Wireless Application Protocol (WAP) technology: GSM, GPRS, and Universal Mobile Telephone Service (UMTS))
- Suitability for a wide range of lightweight terminal types, including next generation mobile phones / PDAs / PC hybrid terminals.

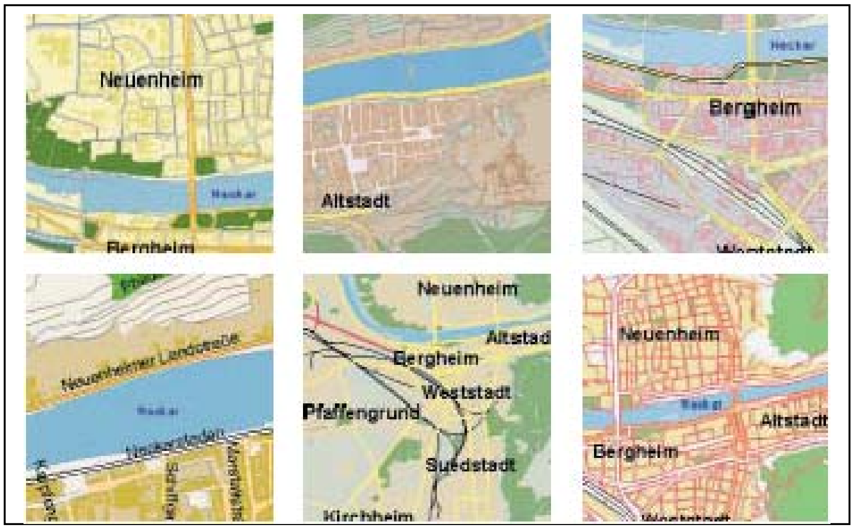
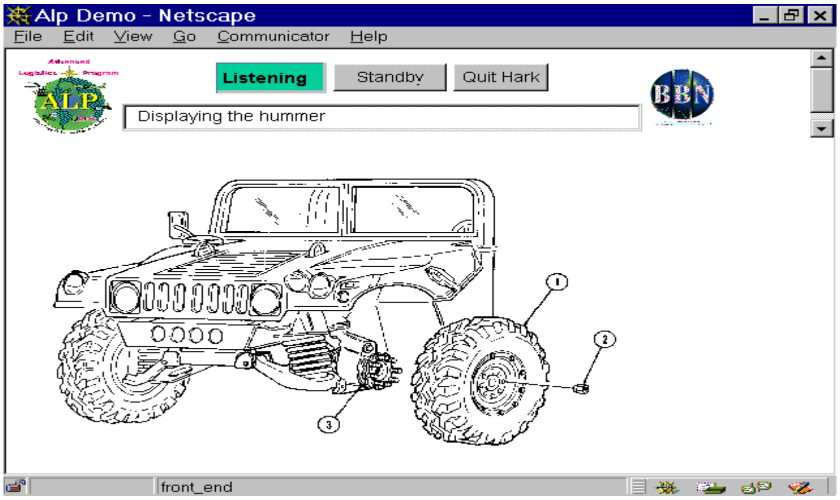


Figure 2.6: Example of CRUMPET's map production

These features will provide end users (tourists) with a user-aware location-based service that utilises GIS (geographical information systems) for the LBS (location based service) aspect, and personalisation techniques to generate a user-adapted interface.

2.2.4 VoiceLog

VoiceLog (BBN 2002, Bers et al. 1998) is a project incorporating logistics, thin clients, speech recognition, OCR (Optical Character Recognition) and portable computing. VoiceLog consists of a slate laptop connected by a wireless 14.4K modem to a server that facilitates speech recognition, exploded views/diagrams of military vehicles and direct connection to logistics. The diagrams of the military vehicles are available via a webpage interface, an example of which is shown in Figure 2.7. The data flow in VoiceLog is depicted in Figure 2.8.



Displaying the hummer

Figure 2.7: VoiceLog’s interface displaying a part-diagram

The idea behind VoiceLog is that a person in the field has support for specifying what is damaged on a vehicle using diagrams, and for ordering the parts needed to repair the vehicle. A sample of an interaction with the system follows (U - input; VL - VoiceLog response):
U: "Show me the humvee." [humvee is a nickname for the HMMWV military jeep]
VL: displays an image of the HMMWV.
U: "Expand the engine."
VL: flashes the area of the engine and replaces the image with a part diagram for the engine.
U: (while pointing at a region with the pen) "Expand this area."
VL: highlights the selected area, the fuel pump, and displays an expanded view.
U: (points to a specific screw in the diagram) "I need 10."
VL: brings up an order form filled out with the part name, part number and quantity fields for the item.

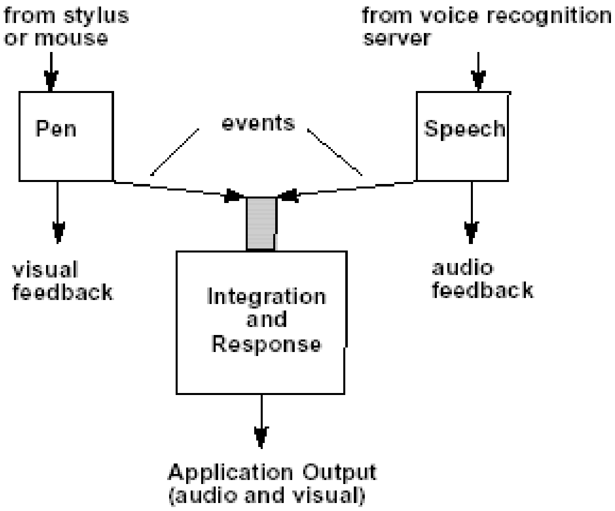


Figure 2.8: Data flow in VoiceLog

The laptop accepts spoken input (recognised on the server) and touch screen pen input. The visual part of the system consists of web pages showing diagrams and order forms which is supported by a program controlling the speech interface. The user of VoiceLog selects items on the display with the pen or with speech and specifies actions verbally. VoiceLog splits the computational workload of speech recognition such that the client machine performs front-end signal processing and sends the vector-quantified data to the server for decoding into words. This design enables VoiceLog to cope with large vocabulary recognition because the computational and memory intensive processing is done on the server.

2.2.5 MUST

MUST (MUltimodal multilingual information Services for small mobile Terminals) (Almeida et al. 2002) is a project which implements a tourist guide to Paris. The aims of MUST are to 1) get a better understanding of the issues that will be important for future multimodal and multilingual services in the mobile networks accessed from small terminals; 2) evaluate Multimodal interaction through testing with naïve non-professional users. MUST uses speech and pen (pointing/haptic deixis) for input, and speech, text, and graphics for output. The architecture of MUST is shown in Figure 2.9.

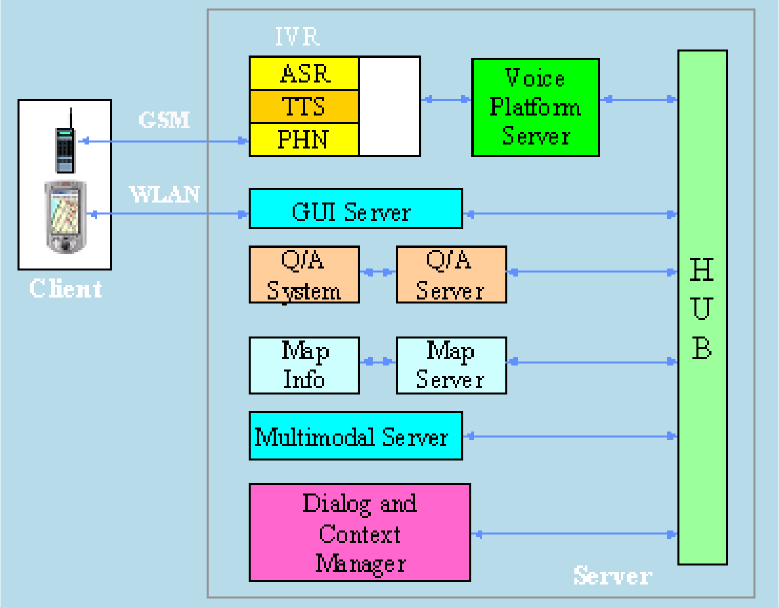
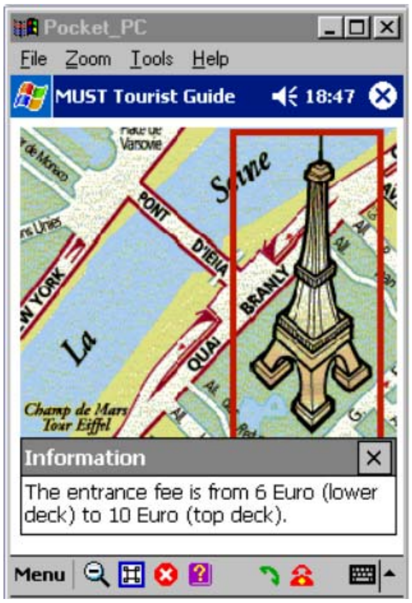


Figure 2.9: MUST system architecture

The server side of the architecture combines a number of specialised modules that exchange information among each other. The server is accessed by the user through a thin client that runs on the mobile terminal (e.g. Compaq iPaq/mobile phone). The voice servers provide an interface to ISDN and PSTN telephony and advanced voice resources such as Automatic Speech Recognition (ASR) and Text-To-Speech synthesis (TTS). The multimodal question-answering (Q/A) system uses a combination of syntactic/semantic parsing and statistical natural language processing techniques to search the web for potentially relevant documents. The GALAXY Communicator Software Infrastructure, a public domain reference version of DARPA Communicator provides the HUB which acts as the underlying inter-module communication framework of the system.

Spoken utterances in MUST are forwarded to a speech recogniser by a telephone module. Text and Pen inputs are transferred from a GUI client via a TCP/IP connection to a GUI server. The inputs from the speech recogniser and the GUI server are then integrated on a multimodal sever, a form of late fusion, and then passed to the Dialogue/context Manager (DM) of the system. Possible inputs to MUST include "What hotels are there near the Notre Dame?", "What restaurants are in this neighborhood?" combined with Haptic input on the screen, or "What is the entrance fee to the Eiffel tour?". An output for the latter example query is displayed in Figure 2.10. The DM interprets the result and acts accordingly, for example by contacting the Map Server connected to the system and fetching information to be presented to the user. The information is then sent to the GUI server and voice server via the multimodal server that performs the fusion (extracting data addressed to each output modality- speech, graphics). In addition, a multilingual Question/Answering system has been integrated into the MUST system to handle out of domain requests.



The entrance fee is from 6 Euro (lower deck) to 10 Euro (top deck).

Figure 2.10: Example output from MUST

The main point to note about the systems described in the previous sections is that current mobile intelligent multimedia systems fail to take into consideration network constraints and especially the bandwidth available when designing the output presentation destined for a mobile device. One framework, RAJA (section 2.2.2), which is aimed at the development of resource-adaptive multi-agent systems is intended to provide resource-aware modules for DEEP MAP but this has not been integrated yet. If the bandwidth available to a device is low then it's obviously inefficient to attempt to use video or animations as the output on the mobile device. This would result in an interface with depreciated quality, effectiveness and user acceptance. This is an important issue as regards the usability of the interface. Learnability, throughput, flexibility and user attitude are the four main concerns affecting the usability of any interface. In the case of the previously mentioned scenario (reduced bandwidth => slower/inefficient output) the throughput of the interface is affected and as a result the user's attitude also. This is only a problem when the required bandwidth for the output modalities exceeds that which is available; hence, the importance of choosing the correct output modality/modalities in relation to available resources.

2.3 Network-adaptive multimedia models

Adaptation to limited resources is the main focus of TeleMorph. Wireless networks, client device and server resources are our main concerns within this issue that affect the quality of media being presented to the end user on a mobile device. Mobile computing suffers from:

Wireless networks problems:

- Varying Bandwidth
- Variable Latency
- Packet losses
- High error rates
- Quality of Service that cant be ensured
- Possibility of disconnection

Mobile device limitations:

- Limited processing power
- Limited battery life
- Limited memory
- Poor display resolution
- Tedious data input

As mobile computing suffers from varying connection and environment conditions mobile applications must be able to adapt to these variations dynamically, in order to provide a truly seamless experience to the end user. A variety of solutions to these problems exist and are now discussed. Some of these include:

- 'Transcoding proxies' incorporating distillation and refinement techniques on the media;
- 'End-to-End' approaches where a number of discrete versions of the media are encoded adhering to multiple fidelities;
- A combination of these two is also possible where the server holds a variety of 'resolutions' of each media. But the server may not maintain content appropriate for all client/network speed combinations so more refinement may be required at the proxy in order to tailor the multimedia into a format more specific to the network and device requirements/constraints.

2.3.1 Transcoding proxies

A considerable amount of work has been done in the area of information access from mobile platforms, using mostly the client-proxy-server model. Clients should not be connected directly to a service as this can force application-specific coupling between the front end and back end inhibiting support for transparent mobility. Proxies solve this by acting as intermediaries between clients and servers, decoupling the data delivery from the server to the client. In the case of mobile hosts, a proxy is often an application that executes in the wired network to support the host. This location is frequently the base station, the machine in the wired network that provides the radio interface. As the user moves, the proxy may also move to remain in the communication path from the mobile device to the fixed network. Logically, the proxy hides the client from the server, which thinks it communicates with a standard client (i.e. a client that executes on a powerful desktop directly connected to the wired network) (Hokimoto & Nakajima 1997). Media content coded to handle the channel characteristics (e.g. error rates) of the wired network need codecs that are better suited to the wireless network. This code transformation is the main function of transcoding proxies. The basic assumption underlying transcoding proxies is that the servers have a document at a fixed "resolution" of a given media, and so the proxy needs to appropriately modify the multimedia content to suit the client side resources and bandwidth. This modification is referred to as 'distillation' or 'refinement' as the proxy is distilling/refining the media in response changing client or network resources. 'Distillation' is defined as highly lossy, data-specific compression that preserves most of the semantic content of the document (Fox et al. 1996). It can be thought of as optimising the object content with respect to representation constraints with the purpose of not only providing a more transmittable object but also allowing the user to evaluate the value of downloading the original version of the object, or some part of it. Table 2.2 lists the "axes" of compression corresponding to three important datatypes: formatted text, images, and video streams.

Semantic type	*Some specific encodings*	*Distillation axes or quality levels*
Image	GIF, JPEG, Png, PPM, PostScript figure	Resolution, colour, depth, colour palette
Text	Plain, HTML, PostScript, PDF	Heavy formatting, simple markup, plain text
Video	MPEG, H.261, H.263, M-JPEG, VQ, NV, Cell-B	Resolution, frame rate, colour depth, progression limit (for progressive encodings)

Table 2.2: Three Datatypes and related distillation axes

There are logical limits as to how severe a degradation of quality is possible before the source object becomes unrecognisable, hense comprimising semantic usefulness. ‘Refinement’ is defined as the process of fetching some part (possibly all) of a source object at increased quality, perhaps even the original representation (Fox et al. 1996). This could be due to user choice or otherwise a dynamic response to an improvement in network or client device conditions. As with distillation, the refinement technique is specific to the semantic type and the implementation of the technique requires intimate knowledge of the encoding.

A considerable amount of work has been done in the area of information access from mobile platforms using the client-proxy-server model. In GloMop (Fox et al. 1996, Fox & Brewer 1996, Katz et al. 1996, Brewer et al. 1998) (part of the Daedalus group at Berkeley) the proxy performs distillation of the document received from the server before sending it to the client. For instance, GloMop performs transcoding of motion JPEG to subsmapled H.261 for video data. The Mowgli system (Liljeberg et al. 1996a,b) consists of two mediators located on the mobile host and the mobile-connection host which use the Mowgli HTTP protocol to communicate with each other, reducing the number of round-trips between the client and server. Mowgli reduces the data transfer over the wireless link in three ways: data compression, caching and intelligent filtering. The notion of web intermediaries to affect transcoding and personalisation related functionalities is also the focus of IBM’s Web Browser Intelligence (WBI) (Barrett et al. 1997) system.

In Odyssey (Noble et al. 1997), the proxy is developed in the context of what the authors term agile, application aware adaptation. Basically, they allow an application to register with the OS its expectations about a resource and the variability that it can tolerate. The Odyssey system monitors resources, and informs the applications via upcalls when the resource value strays outside the bounds decided by the application. The application can then adapt its behaviours. For web browsing in particular, a module called Cellophane on the client transforms HTTP requests into file operations on Odyssey web objects and selects fidelity levels for images which are forwarded to a distillation server. However this approach is specific to the Odyssey file system. Other lines of work have sought to support disconnected operation using local caching. Several commercial offline browsers download documents into a local disk for later viewing. The user specifies a set of links, similar to a bookmarks file, which the application downloads and caches. Some of these run a proxy web server on the portable machine to deliver cached documents. One common problem of these approaches is that they propose to conserve one scarce resource (connection bandwidth) in the mobile scenario by using up another scarce resource (disk space).

2.3.2 End-to-End approach

Seshan et al. (1997) were one of the first to present the notion that networks could use network performance parameters to download documents from a server at different “fidelities”, and explicitly mentioned that this was something beyond text-only pages.

Implicit in their approach was the idea that the server would indeed have different fidelities to present to the client. With disk storage increasingly inexpensive, it is more realistic to expect that content providers will react to the market and make available multimedia content in multiple fixed resolutions. With an end-to-end approach the server can carry out more complex transformations (e.g. summarising video/text, converting text to audio, identifying keyframes in a video etc.) based on client device or network needs. These are computed offline and stored. Most of these computations are too compute intensive to be done on-the-fly at the proxy, and so are not possible in a proxy only approach. The end-to-end model is predicated upon the client being able to "identify" itself to the server. There has been a very recent initiative, CC/PP (Composite Capability/Preference profiles) (CCPP 2003), to use Resource Description Framework (RDF) -based HTTP extensions as a standard mechanism for this.

Pfisterer (2001) describes a system that is based on the end-to-end model utilising the RAJA framework previously discussed in section 2.2.2. Pfisterer discusses "Resource-Aware Multi-Fidelity Video Streaming" where the streaming server can dynamically vary the quality of streamed video content. This was achieved by storing pre-encoded videos at different data rates and switching between them in real time. The system transmitted data at rates ranging from 100K bps to 200K bps. On the receiver side, the received data is monitored and compared to the required data rate of the streamed video. This data is then used by the client agent as basis as a for adaptation decisions.

The end-to-end model has also been utilised by companies designing intelligent streaming technologies (e.g. Real Networks' Real Media (RealNetworks MediaServer 2003) and Microsoft's Windows Media (Windows MediaServer 2003)). Real Media and Windows Media handle the media thinning issue by encoding several different bitrates into a single file, thereby giving the content producer the ability to raise or lower the bitrate dynamically and seamlessly to accommodate changing network conditions during playback. Content must be encoded as multiple-bit-rate stream. In multiple-bit-rate encoding, a number of discrete, user-definable audio and video streams are encoded into a single media stream. The streams are encoded from the same content, but each is encoded at a different bit rate. When the client connects to the server to receive a multiple-bit-rate file or broadcast stream, the server only sends the set of audio and video streams that is most appropriate for the current bandwidth conditions. The process of selecting the appropriate stream is completely transparent to the user. This model of media streaming is referred to as intelligent streaming.

2.3.3 Transcoding proxy & End-to-End combination

Since it may not be feasible for every content server to provide and maintain data in a format and resolution appropriate for all possible client/network combinations, it may be more viable to utilise some sort of transcoding proxy and end-to-end combination. With this model the server would make the content available in a variety of resolutions (possibly 4 or 5 depending on the frequency of access and on the client device type accessing the information). Then further media distillation would be performed at the transcoding proxy to provide additional content adjustments to suit device/network constraints.

The system proposed by Joshi (2000) implements a combined end-to-end and proxy based approach. The system consists of a proxy-based mobile browser called Mowser and a web intelligent query system (W3IQ). Mowser is a transcoding proxy to handle multimedia content and W3IQ is a proxy that provides for personalisation and asynchronous operation. In the case that the content server is not able to provide multimedia at a resolution appropriate for the client device and the connecting wireless network, then the proxy monitors and transcodes the server response using preferences set by the mobile client.

2.3.4 Mobile/Nomadic computing

Nomadic routers, agents and gateways are alternative interposition strategies for multimedia adaptation to those already discussed. Kleinrock (1995) describes a "nomadic router" that is interposed between an end system and the network. This module observes and adapts to the means by which the end system is connected to the network, e.g., through a phone line in a hotel room versus through the LAN in the home office. It might decide to perform more file caching or link compression when the end system is connected through a low bandwidth link and/or invoke additional security, such as encryption, when operating away from the home office. Similarly, nomadic agents and gateways (Kleinrock 1995) are nodes that support mobility. They are located at strategic points that bridge networks with vastly different bandwidth and reliability characteristics, such as the junctions between wired and wireless networks. Application-specific services performed at gateways include file caching and the transcoding of images (Amir et al. 1995).

2.3.5 Active networks

In an active network (Tennenhouse & Wetherall 1996, Tennenhouse et al. 1997) the routers/switches/nodes of the network perform customised computations on, and modify, the contents of packets being transmitted. This process can be customised on a per user or per application basis. In contrast, the role of computation within traditional packet networks, such as the Internet, is extremely limited. Although routers may modify a packet's header, they pass the user data opaquely without examination or modification. The end-user can also inject programs into the network, thereby tailoring the node processing to be user- and application-specific. The main advantages of active networks are that:

- Unlike in conventional networks, the nodes are not dormant to data processing, and hence their latent computational potential is utilised owing to a much better service throughput at destination.

Portability of services across domains using highly adaptive protocols is maximised, yielding richer interactions and hence obviating the necessity for the exchange of fixed data formats.

In this section various network-adaptive multimedia models have been discussed, including transcoding proxies, the end-to-end approach, a combination of these two, nomadic computing, and active networks. These techniques are used in multimedia streaming architectures where the multimedia is too rich in content and therefore too large in size to be transmitted effectively across mobile networks due to their limited speed and also the limitations that exist on the mobile client device receiving the data. To resolve this issue, a variety of techniques were proposed from distilling the media content (to lower quality and hence smaller file sizes), to providing multiple fidelities of the media (pre-coded to various resolutions and stored), to performing customised modifications to the media at active network nodes. These solutions to resource deficiencies are focused on multimedia files that are prefabricated. TeleMorph differs from this in that the multimedia streamed will be part of a multimodal presentation that will be designed on the fly/dynamically in relation to available bandwidth. All the same, these techniques provide a good insight into resource-aware media adaptation.

2.4 Semantic representation

One of the key research problems that exists in intelligent multimedia presentation is semantic representation. Semantic representation relates to the method employed to represent the meaning of media information (Romary 2001, Bunt & Romary 2002, Reithinger et al. 2002, Mc Kevitt 2003). A multimodal semantic representation must support -

- Both interpretation and generation,
- Any kind of multimodal input and output,
- A variety of semantic theories.

A multimodal representation may contain architectural, environmental, and interactional information. Architectural representation indicates producer/consumer of the information, information confidence, and input/output devices. Environmental representation indicates timestamps and spatial information. Interactional representation indicates speaker/user's state. The two traditional semantic representations are XML and Frames. Figure 2.11 illustrates the relationship between multimodal semantic representations and intermediate-level semantic representations for language and visual information. Multimodal semantic representations are media-independent and are usually used for media fusion, integration and coordination. Output semantic representations are media dependent (visual/language) and are typically used for media realisation.

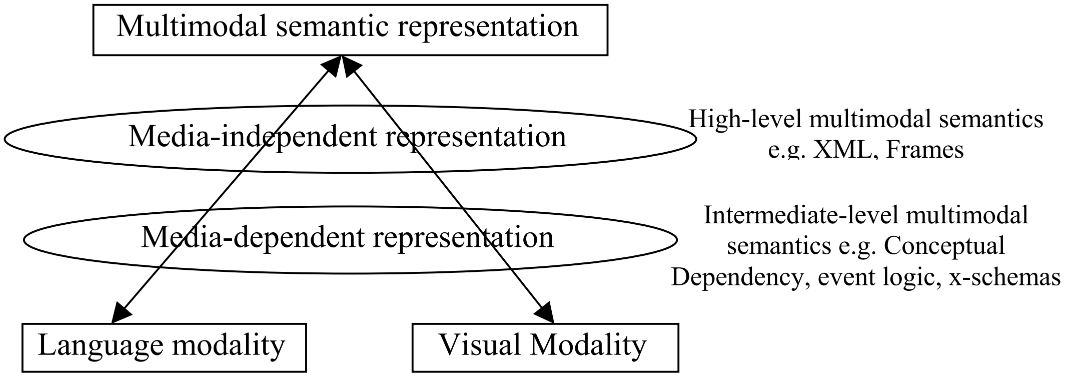

Multimodal semantic representation
Media-independent representation
Media-dependent representation
Language modality
Visual Modality
High-level multimodal semantics e.g. XML, Frames
Intermediate-level multimodal semantics e.g. Conceptual Dependency, event logic, x-schemas

Figure 2.11 Multimodal semantic representation

2.4.1 Frames

Frames were first introduced by Minsky (1975) to semantically represent situations in order to permit reasoning and decision making. A frame is a collection of attributes or slots and associated values that describe some real world entity. Frames are based on a psychological view of human memory and the basic idea is that on meeting a new problem humans select an existing frame (a remembered framework) to be adapted to fit new situations by changing appropriate details. Frames on their own are not particularly helpful but frame systems are a powerful way of encoding information to support reasoning. Set theory provides a good basis for understanding frame systems. Each frame represents a class (set), or an instance (an element of a class). Examples of Frames can be seen using examples from CHAMELEON (Brøndsted et al. 1998, 2001) in Figure 2.12. Intelligent multimedia applications using Frames to represent multimodal semantics include:

- CHAMELEON (Brøndsted et al. 1998, 2001)
- AESOPWORLD (Okada 1996)
- REA (Cassell et al. 2000)
- Ymir (Thórisson 1996)
- WordsEye (Coyne & Sproat 2001)

CHAMELEON is a distributed architecture of communicating agent modules processing inputs and outputs from different modalities and each of which can be tailored to a number of application domains. An initial application of CHAMELEON is the IntelliMedia WorkBench. IntelliMedia focuses on computer processing and understanding of signal and symbol input from at least speech, text and visual images in terms of semantic representations. CHAMELEON is a software and hardware platform tailored to conducting IntelliMedia in various application domains. In the IntelliMedia WorkBench a user can ask for information about things on a physical table. Its current domain is a Campus Information System where 2D building plans are placed on the table and the system provides information about tenants, rooms and routes and can answer questions like *"Where is Paul's room?"* in real time. Figure 2.12 shows an example of the Frame semantic representation that is used in CHAMELEON.

Input Frames	Output Frames
[MODULE INPUT: input INTENTION: intention-type TIME: timestamp]	[MODULE INTENTION: intention-type OUTPUT: output TIME: timestamp]
[SPEECH-RECOGNISER UTTERANCE:(Point to Hanne's office) INTENTION: instruction! TIME: timestamp]	[SPEECH-SYNTHESISER INTENTION: declarative UTTERANCE: (This is Paul's office) TIME: timestamp]
[GESTURE GESTURE: coordinates (3, 2) INTENTION: pointing TIME: timestamp]	[LASER INTENTION: description (pointing) LOCATION: coordinates (5, 2) TIME: timestamp]

Figure 2.12 Examples of frame semantic representation in CHAMELEON

Frames are coded in CHAMELEON with messages built as predicate-argument structures following a specific BNF definition (Brøndsted et al. 1998). Frames represent some crucial elements such as module, input/output, intention, location, and timestamp. Module is simply the name of the module producing the frame (e.g. NLP). Inputs are the inputs recognised whether spoken (e.g. "Show me Paul's office") or gestures (e.g. pointing coordinates) and outputs the intended output whether spoken (e.g. "This is Paul 's office.") or gestures (e.g. pointing coordinates). Timestamps can include the times a given module commenced and terminated processing and the time a frame was written on the blackboard. The frame semantics also includes representations for two key phenomena in language/vision integration: reference and spatial relations. Frames can be grouped into three categories: (1) input, (2) output and (3) integration. Input frames are those which come from modules processing perceptual input, output frames are those produced by modules generating system output and integration frames are integrated meaning representations constructed over the course of a dialogue (i.e. all other frames).

2.4.2 XML

XML (eXtensible Markup Language) specification was published as a W3C (World Wide Web Consortium) recommendation (W3C 2003). As a restricted form of SGML (the Standard Generalised Markup Language), XML meets the requirements of large-scale web content providers for industry-specific markup, data exchange, media-independent publishing, workflow management in collaborative authoring environments, and the processing of web documents by intelligent clients. Its primary purpose is as an electronic publishing and data interchange format. XML documents are made up of entities which contain either parsed or unparsed data. Parsed data is either markup or character data (data bracketed in a pair of start and end markups). Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

Unlike html, XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. A software module which is used to read XML documents and provide access to their content and structure is called an XML processor or an XML parser. It is assumed that an XML processor is doing its work on behalf of an application. Any programming language such as Java can be used to output data from any source in XML format. There is a large body of middleware written in Java and other languages for managing data either in XML or with XML output. Intelligent multimedia systems using XML to represent their multimodal semantics include:

- SmartKom (Wahlster 2003) uses M3L (MultiModal Markup Language), an XML derivative.
- BEAT (Cassell et al. 2001)
- SAM (Cassell et al. 2000)
- MIAMM (Reithinger et al. 2002) utilises MMIL (MultiModal Interface Language), an XML derivative.
- MUST (Almeida et al. 2002) uses MXML (MUST XML), an XML derivative.
- IMPROVISE (Zhou & Feiner 1997, 1998).

In SMARTKOM, M3L (MultiModal Markup Language), an XML derivative is used. M3L is a complete XML language that covers all data interfaces within the complex multimodal dialog system. Instead of using several quite different XML languages for the various data pools, an integrated and coherent language specification was developed, which includes all sub-structures that may occur. In order to make the specification process manageable and to provide a thematic organization, the M3L language definition has been decomposed into about 40 schema specifications. The basic data flow from user input to system output continuously adds further processing results so that the representational structure is refined, step-by-step. Figure 2.13 shows an example of the semantic representation language M3L that is used in SmartKom.

```
<presentationTask>
  <presentationGoal>
    <inform> <informFocus> <RealizationType>list </RealizationType> </informFocus> </inform>
    <abstractPresentationContent>
         <discourseTopic> <goal>epg_browse</goal> </discourseTopic>
         <informationSearch id="dim24"><tvProgram id="dim23">
           <broadcast><timeDeictic id="dim16">now</timeDeictic>
              <between>2003-03-20T19:42:32 2003-03-20T22:00:00</between>
              <channel><channel id="dim13"/> </channel>
           </broadcast></tvProgram>
         </informationSearch>
      <result> <event>
         <pieceOfInformation>
           <tvProgram id="ap_3">
                  <broadcast> <beginTime>2003-03-20T19:50:00</beginTime>
                              <endTime>2003-03-20T19:55:00</endTime>
                              <avMedium> <title>Today's Stock News</title></avMedium>
                              <channel>ARD</channel>
                  </broadcast>……..
       </event>   </result>
</presentationGoal> </presentationTask>
```

Figure 2.13 Example of SmartKom's M3L Semantic Representation language

The M3L code represents the modality-free presentation goal that is transformed into a multimodal presentation by SmartKom's media fission component and unimodal generators and renderers. The presentation goal shown in Figure 2.13 is coded in M3L and indicates that a list of TV broadcasts should be presented to the user in response to a user-request.

2.4.3 SMIL

The Synchronised Multimedia Integration Language (SMIL) (Rutledge 2001, Rutledge & Schmitz 2001, SMIL 2003a, 2003b) is an XML based semantic representation language for encoding multimedia presentations for delivery over the web. The World Wide Web Consortium (W3C 2003) developed and first released the specification in 1998. SMIL has been installed in over 200 million desktops world wide, primarily because of its integration into RealPlayer (2003), Quicktime (Apple 2003) and Internet Explorer (Microsoft 2003). Several vendors have released SMIL 2.0 language profile players and tools, including Oratrix (2003) and RealNetworks (2003). The Third Generation Partnership Project (3GPP) (Elsen et al. 2003) consortium has used the SMIL basic profile as the basis for the wireless multimedia specification. In 2001 the W3C updated the SMIL specification with SMIL 2.0. SMIL is basically a collection of XML elements and attributes that can be used to describe the temporal and spatial coordination of one or more media objects. SMIL's focus isn't on automatic data encoding but on integrating and syncronising independent media, thus, it may be necessary to extend the SMIL language to include more semantic information, possibly served by the XML-based MPEG-7 standard discussed in the next section.

The SMIL 2.0 specification defines approximately 50 modules, which are grouped into 10 major functional areas: animation, content control, layout, linking, media objects, metainformation, structure, timing and synchronization, time manipulation, transitions. These functional groupings represent collections of individual SMIL 2.0 modules, each of which defines elements and attributes intended to address specific multimedia issues in a reusable manner. The number of modules per functional grouping varies from 2 to about 20. Generally, the more modules per grouping the finer each module's granularity. Figure 2.14 below shows the functional groupings.

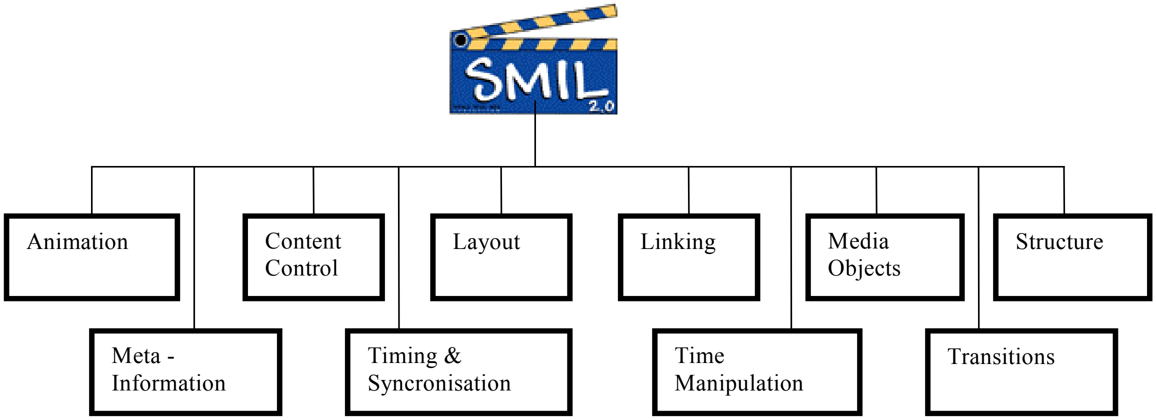


Figure 2.14 SMIL 2.0's functional grouping of module sets

Of these functional groups the timing and syncronisation functional group represents the core of the SMIL specification. The group consists of 19 modules, each of which defines a collection of XML elements and attributes that control some timing aspect. The three basic SMIL 2.0 timing elements are seq, par, and excl. Each of these elements form parent timing containers in which one can place media objects or other timing containers. Section 2.5.1 will discuss seq, par and excl in further detail in relation to syncronisation of modalities in a presentation. One of the major innovations of SMIL is support for conditional content control via the switch element given in Figure 2.15:

```
<switch>
  <video src=“...”    systemBitrate=“115200”/>
  <seq systemBitrate=“57344”>
        <img src=“img1.png” dur=“5s”/>
        ...
        <img src=“img12.png” dur=“9s”/>
  </seq>
  <text src=“desc.html” dur=“30s”/>
</switch>
```

Figure 2.15 Example of content control in SMIL

In the fragment in Figure 2.15, a video object is rendered if the system bit rate (actual or configured) is set at 112 Kbytes or above. If the bit rate is 56 Kbytes or above, but below 112 Kbytes, a player shows a sequence of images instead. If no other element has satisfied the pre-requisites for selection in the switch, the SMIL player defaults to the text object.

One of the SMIL 1.0 switch element’s limitations was that it only allowed selection based on a predefined list of system attributes. SMIL 2.0 extends this notion with a user-defined set of test attributes: custom test attributes. The author can define the custom test attributes, and the user can select them directly or indirectly via the player. SMIL 2.0 also lets test attributes be used inline - that is, outside the switch element. Any media object reference containing a system or custom test attribute will be evaluated as if it were wrapped into a single-element switch. The content control facilities provide a means for dynamically selecting objects in a presentation, but this must be coded into the SMIL by the author of the presentation. With this system for constraint adaptation only those test attributes specifically identified by the author are considered when adapting the output. Whilst this mechanism allows for a sort of dynamic behavior within the code, the declarative nature of the test metrics restrict the potential for on-the-fly identification and adaptation to various constraints. Also, unless the author exhausts all combinations of media sources there is the possibility that all available resources will not be exploited to full potential.

For example, in Figure 2.15 if the bit rate was 110KBs this would be just short of the pre-requisite for presenting the video source, but instead media is displayed which is suited to a bit rate of 56KBs. This is an example of the lack of intuitive awareness of available resources. The only way around this as stated is to exhaust all possible bit rate values and declare the appropriate media for each which would likely lead to an impractical amount of coding for the author. The presentation adaptation process needs to occur at run-time based on exact constraint calculations, resulting in a more informed modality selection process.

SMIL 2.0's developers intended for its modularisation to facilitate the reuse of SMIL functionality in other XML languages and help define a number of SMIL profiles. Each profile provides a collection of module sets/functional groupings that users can customise to its primary goal. Figure 2.16 shows the structure of module sets in SMIL, which combine to form the basic profiles to server specific domains.

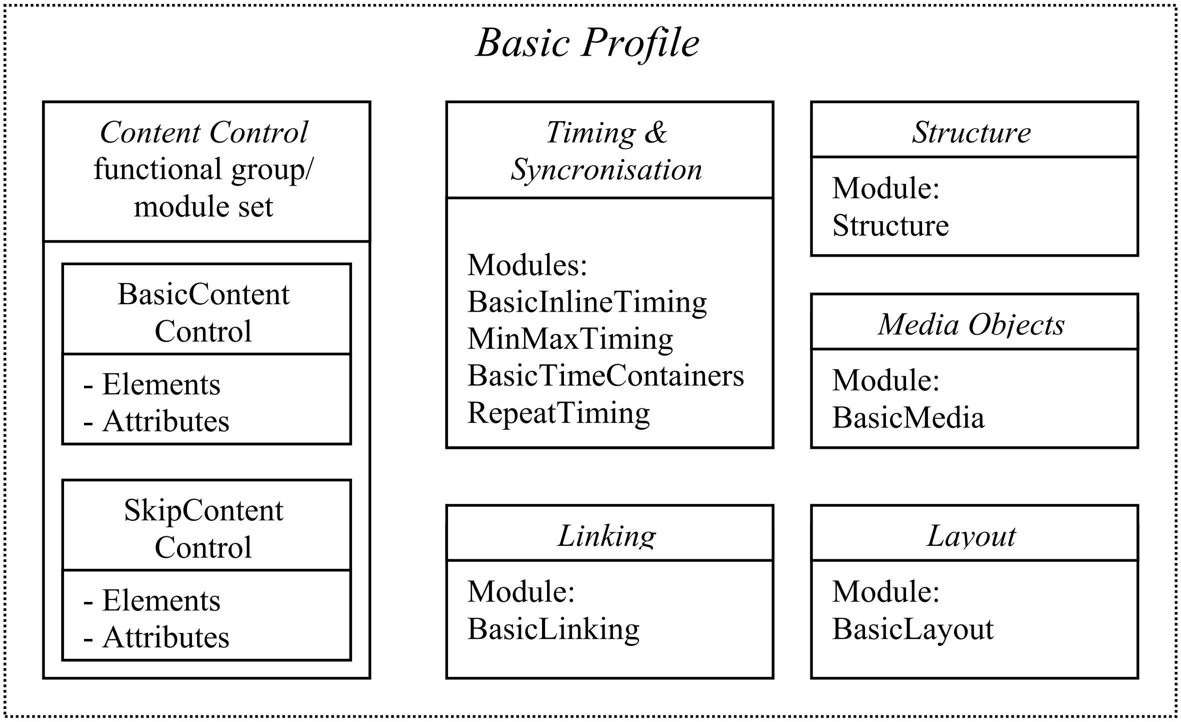


Figure 2.16 Structure of the Basic profile in SMIL

The SMIL 2.0 basic profile is a collection of module sets that supports SMIL on minimal devices, such as mobile phones or Personal Digital Assistants (PDAs). SMIL basic supports the lowest complexity modules of each of the SMIL 2.0 functional groups. The basic profile defines the minimum baseline for documents that can still be considered members of the SMIL language family; individual implementations may add other parts of SMIL 2.0 functionality if appropriate to their devices.

2.4.4 MPEG-7

MPEG-7 (MPEG7 2003), formally named "Multimedia Content Description Interface", is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). The goal of the MPEG-7 standard is to allow interoperable searching, indexing, filtering, and access of audio-visual (AV) content by enabling interoperability among devices and applications that deal with AV content description. It is not intended to replace MPEG-1, MPEG-2 or MPEG-4 but instead to provide complimentary functionality to these standards: representing information about the content and not the content itself. MPEG-7 describes specific features of AV content, as well as information related to AV content management. MPEG-7 descriptions take two possible forms: 1) a textual XML form suitable for editing, searching, and filtering, and 2) a binary form suitable for storage, transmission, and streaming delivery. As MPEG-7 represents information about AV content, it could be integrated with SMIL (as both are XML based) to create a comprehensive multimodal semantic representation language. Overall, the MPEG-7 standard specifies four types of normative elements:

- Descriptors
- Description Schemes (DSs)
- Description Definition Language (DDL)
- Coding schemes

The MPEG-7 Descriptors are designed primarily to describe low-level audio or visual features such as color, texture, motion, audio energy, etc., as well as attributes of AV content such as location, time, quality, etc. It is expected that most Descriptors for low-level features shall be extracted automatically in applications. On the other hand, the MPEG-7 DSs are designed primarily to describe higher level AV features such as regions, segments, objects, events, and other immutable metadata related to creation and production, usage, and so forth. The DSs produce more complex descriptions by integrating together multiple Descriptors and DSs, and by declaring relationships among the description components. In MPEG-7, the DSs are categorised as pertaining to the multimedia, audio, or visual domain. Typically, the MDSs describe content consisting of a combination of audio, visual data, and possibly textual data, whereas the audio or visual DSs refer specifically to features unique to the audio or visual domain, respectively. In some cases, automatic tools can be used for instantiating the DSs, but in many cases instantiating DSs requires human assisted extraction or authoring tools.

The MPEG-7 DDL is a language for specifying the syntax of the DSs and Descriptors. The DDL also allows the MPEG-7 standardised DSs and Descriptors to be extended for specialised applications. The DDL is based on the W3C XML Schema Language (W3C XML 2003). An MPEG-7 description is produced for a particular piece of AV content by instantiating the MPEG-7 DSs or Descriptors as defined by the DDL. The MPEG-7 coding schemes produce a binary form of description that is compact, easy to stream, and resilient to errors during transmission. These multimedia description schemes can be organised into five functional areas:

- Basic Elements
- Content Management and Description
- Navigation and Access
- Content Organisation
- User Interaction

Figure 2.17 gives an overview of how each of these areas relate in MPEG-7.

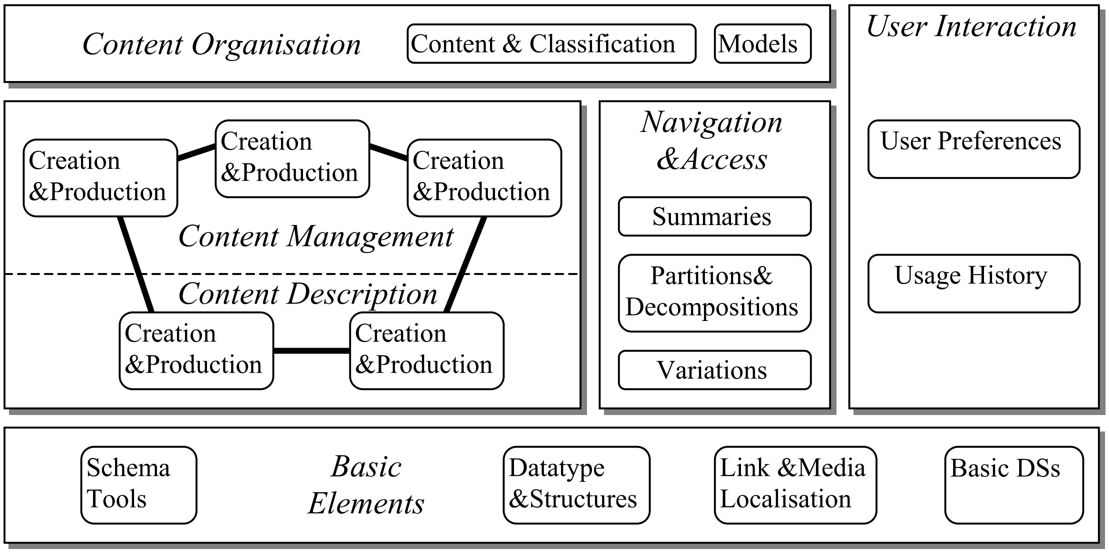


Figure 2.17 MPEG-7 overview

Thus, the Multimedia Content Description Interface (MPEG-7) standard only specifies the format for descriptions of content, and not the algorithms to utilise this description. It is anticipated that MPEG-7 will make the web more searchable for multimedia content than it currently is for text. Some examples of applications that could benefit from MPEG-7 technology include digital libraries, multimedia directory services, broadcast media selection and multimedia editing applications.

2.4.5 Semantic web

The Semantic Web (Berners-Lee et al. 2001) is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. Research on the Semantic Web is driven by the need for a new knowledge representation framework to cope with the explosion of unstructured digital information on the existing Web. Current Semantic Web research focuses on the development of ontology languages and tools for constructing digital information that can be “understood” by computers. The origin of the Semantic Web research goes deep in the roots of Artificial Intelligence research (e.g., knowledge representation and ontology). Recent Semantic Web research begins with the DAML (DARPA Agent Markup Language) (Zaychik 2001, Horrocks 2002, Horrocks et al. 2002, DAML 2003) effort in the US and the OIL (Ontology Inference Layer) (Horrocks 2002, Horrocks et al. 2000, 2002) effort in Europe. The original DAML language was then combined with many of the ontology modeling features from the OIL language, and the result is the DAML+OIL language. In late 2001, the World Wide Web Consortium (W3C) established the Web Ontology Working Group with the goal of introducing Semantic Web technologies to the main stream web community. The group has specified a language OWL (Ontology Web Language) (OWL 2003) that is based on DAML+OIL and shares many of its features (e.g., using RDF as the modeling language to define ontological vocabularies and using XML as the surface syntax for representing information).

OWL is a component of the W3C Semantic Web activity. It is designed for use by applications that need to process the content of information, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL facilitates greater machine interpretability of Web content than that supported by XML, Resource Description Framework (RDF), and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL is a revision of the DAML+OIL web ontology language incorporating lessons learned from the design and application of DAML+OIL. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL(description logics), and OWL Full.

2.4.6 Speech markup language specifications

There is an emerging interest in combining Multimodal interaction with natural language processing for Internet access. Some standards of XML (eXtensible Markup Language) already exist that are specifically designed for the purpose of Multimodal access to the Internet. SALT (2003) and VoiceXML (2003) are both markup languages for writing applications that use voice input and/or output. Both languages were developed by industry consortia (SALT Forum and VoiceXML Forum, respectively), and both contribute to W3C as part of their ongoing work on speech standards. The reason there are two standards is mainly because they were designed to address different needs, and they were designed at different stages in the life cycle of the Web. VoiceXML and SALT are relevant to TeleMorph because they are standards that are striving towards enabling access to information and services through a respective multimodal interface, something TeleMorph aims to do using speech, text and haptic deixis input modalities, and speech, graphics and text output modalities.

VoiceXML

VoiceXML (2003) was announced by AT&T, Lucent and Motorola. It arose out of a need to define a markup language for over-the-telephone dialogs- Interactive Voice Response (IVR) applications. VoiceXML is designed for creating audio dialogs that feature synthesised speech, digitised audio, recognition of spoken and DTMF (Dual Tone Multi-Frequency) key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response applications.

SALT

SALT (Speech Application Language Tags) (2003) rose out of the need to enable speech across a wider range of devices, from telephones to PDAs to desktop PCs, and to allow telephony (voice-only) and multimodal (voice and visual) dialogs. SALT was founded by Cisco, Comverse, Intel, Philips, Microsoft and SpeechWorks. It's an open standard designed to augment existing XML-based markup languages. The SALT forum has announced that its multimodal access "will enable users to interact with an application in a variety of ways: they will be able to input data using speech, a keyboard, keypad, mouse and/or stylus, and produce data as synthesized speech, audio, plain text, motion video, and graphics. Each of these modes will be able to be used independently or concurrently." (SALT 2002, p.1)

In this section various semantic representation methods have been discussed including frames, XML, SMIL, MPEG-7. Details of these methods were given explaining for each-their intended purpose, and also explaining the syntax of the language. SMIL was described and it was noted that in order for it to represent semantics effectively it may need to be integrated with MPEG-7. OWL, VoiceXML and SALT also featured in this section in discussions on the semantic web and speech markup languages.

2.5 Fusion, integration and coordination of modalities

One of the key research problems that exists in intelligent multimedia presentation is fusion, integration and coordination of modalities. As mentioned previously, the generation process of a multimedia presentation system can be divided into the following processes: the determination of communicative intent, content selection, structuring and ordering, allocation to particular media, realisation in specific media, coordination across media, and layout design. In this section focus is placed on the fusion, integration and coordination of multimedia because an optimal exploitation of different media requires a presentation system to decide carefully when to use one medium in place of another and how to integrate different media in a consistent and coherent manner. In other words, how should the presentation be arranged, in space and in time?

An example of input synchronisation in CHAMELEON (Brøndsted et al. 2001), is where the user may ask a question– "Whose office is this?" The system will then expect a reference to a certain office possibly by gesture; the problem is identifying how long the system should wait to receive this input. CHAMELEON's output modalities must be synchronised in order to portray information effectively to the end user. For example, in response to the query "Where is Paul's office?" a beam of laser light would identify Paul's office, in conjunction with synchronising this with speech output to signify that "This is Paul's office". An important issue when considering coordination and synchronisation between output modalities is the time threshold between modalities. A multimodal presentation consisting of speech and vision needs to be synchronised in order to make sense to the user. Similar to CHAMELEON, an example of output synchronization in REA (Cassell et al. 2000) can be found where an animated agent acts as a real estate agent showing users the features of various models of houses that appear onscreen behind it. If the agent is speaking about a specific feature such as a wall it would be ineffective for the agent to point to this wall after it was finished speaking about the object, hence the importance of syncronisation. A discussion of how SMIL lends itself to fusion, integration and synchronisation of modalities follows in the next section.

2.5.1 SMIL modality synchronisation and timing

Of the ten major functional groups in SMIL 2.0 (see section 2.4.3), the timing and syncronisation functional group represents the core of the specification. The group consists of 19 modules, each of which defines a collection of XML elements and attributes that control some timing aspect. The three basic SMIL 2.0 timing elements are *seq*, *par*, and *excl*. Each of these elements form parent timing containers in which one can place media objects or other timing containers.

SMIL timelines

A SMIL presentation is a structured composition of autonomous media objects. As Figure 2.18 shows, we can use three basic timing containers in a SMIL presentation:

- seq (sequential time container) - A seq container's children are rendered so that a successor child can't begin before its predecessor child completes. A successor child might have an additional start delay, but this delay can't resolve to be negative in relation to the end time of its predecessor. (See Figure 2.18a).
- par (parallel time container) - A par container's children are all rendered in parallel. In terms of a SMIL's timing model, this doesn't mean that they get rendered at the same time but that they share a common timebase defined by the par container, and any or all of the children can be active at any time that the parent par is active. The par is the most general SMIL time container. (See Figure 2.18b).
- excl (exclusive time container) - Only one of the children of an excl can be active at a time. The children's activation order depends on the begin attribute of the excl's children. Typically, each will have an event-based start condition (such as begin="button1.activateEvent") that lets one of the children start on demand. The excl container is new in SMIL 2.0. (See Figure 2.18c).

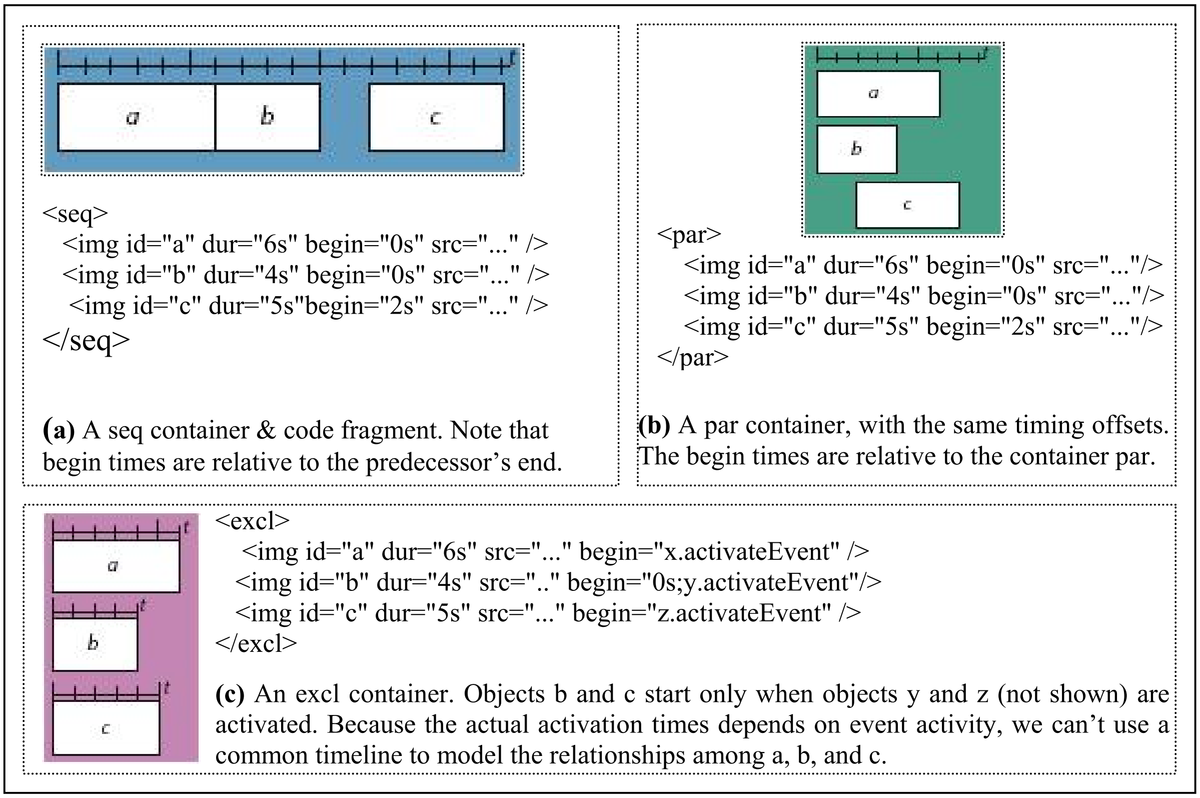
These three basic SMIL 2.0 timing elements (seq, par, and excl) form parent timing containers in which we can place media objects or other timing containers.

SMIL synchronisation

The SMIL timing and synchronisation group consists of 19 modules, each of which defines a collection of XML elements and attributes that control some timing aspect. Table A.1 in A defines each of these modules explaining what purpose they serve in SMIL synchronisation. The SMIL 2.0 specification also provides three high-level synchronisation control attributes which assist with synchronisation behaviour:

- syncBehavior lets a presentation define whether there can be slippage in implementing the presentation's composite timeline,
- syncTolerance defines how much slip is allowed,

- and syncMaster lets a particular element become the master timebase against which all others are measured.



```
<seq>
  <img id="a" dur="6s" begin="0s" src="..." />
  <img id="b" dur="4s" begin="0s" src="..." />
   <img id="c" dur="5s"begin="2s" src="..." />
</seq>
```

(a) A seq container & code fragment. Note that begin times are relative to the predecessor's end.

```
<par>
   <img id="a" dur="6s" begin="0s" src="..."/>
   <img id="b" dur="4s" begin="0s" src="..."/>
   <img id="c" dur="5s" begin="2s" src="..."/>
</par>
```

(b) A par container, with the same timing offsets. The begin times are relative to the container par.

```
<excl>
   <img id="a" dur="6s" src="..." begin="x.activateEvent" />
  <img id="b" dur="4s" src=".." begin="0s;y.activateEvent"/>
  <img id="c" dur="5s" src="..." begin="z.activateEvent" />
</excl>
```

(c) An excl container. Objects b and c start only when objects y and z (not shown) are activated. Because the actual activation times depends on event activity, we can't use a common timeline to model the relationships among a, b, and c.

Figure 2.18 SMIL 2.0 time containers

As stated initially the key issue when considering synchronisation between modalities is the time threshold between modalities and controlling their temporal placement in the output presentation. In this section we have discussed this issue regards current IntelliMedia systems (CHAMELEON and REA) and also in the context of the Synchronised Multimedia Integration Language.

2.6 Intelligent multimedia presentation systems

Intelligent multimedia presentation does not just consist of merging output fragments, but requires a fine-grained coordination of communication media and modalities. Furthermore, in the vast majority of non-trivial applications the information needs will vary from user to user and from domain to domain. Since manually designed presentations must be authored in advance, the ways in which it can be customised for an individual user or situation are limited to those built in by the author. Also, multimedia authoring systems require authors to possess more skills than do single-medium authoring systems. Not only must authors be skilled in the conventions of each medium, but they must also be able to coordinate multimedia in a coherent presentation, determining where and when to use different media, and referencing material in one medium from another. An intelligent multimedia presentation system should be able to flexibly generate various presentations to meet individual requirements of users, situations, and domains. Mapping from semantic representation to an output presentation involves processing the media-independent multimodal semantic representation of a system to produce a media dependent (video/image/text/audio) output presentation. According to Maybury (1994), Maybury and Wahlster (1998) the generation process of a multimedia presentation system can be divided into several co-constraining processes:

- the determination of communicative intent
- content selection
- structuring and ordering
- allocation to particular media
- realisation in graphic, text, and/or other specific media
- coordination across media
- and layout design

It requires intelligent multimedia systems to have the ability of reasoning, planning, and generation. Research in this area initiated during the mid 1990s (Maybury 1993, 1995, Maybury and Wahlster 1998, Mc Kevitt 1995a,b, 1996a,b). This point holds an increasing amount of relevance and importance regarding situations and domains when applied to multimedia presentation on a mobile device as the physical environment, context and device characteristics (e.g. display resolution/screen size/etc.) are subject to variation and must be taken into account appropriately. The relative importance of the individual interface modalities for mobile systems differs from stationary applications: the significance of the natural language interface increases while that of traditional graphical user interfaces decreases. This is a result of the fact that mobile users do not wish to have their visual attention continuously distracted while driving or walking.

Several systems have been developed that automatically generate coordinated multimedia presentations. These systems automate the transition between multimodal semantic representation and multimodal output presentation. Descriptions of some of these systems follow.

2.6.1 COMET

COMET (COordinated Multimedia Explanation Testbed) (Feiner & McKeown 1991a, b), is in the field of maintenance and repair of military radio receiver-transmitters. It is a knowledge-based system, which produces coordinated, interactive explanations that combine text and three-dimensional graphics, all of which are generated on the fly. An example input to COMET is “How do I load the transmission frequency into channel 1?”, resulting in the following system response: “Set the channel knob to position 1. Set the MODE know to SC. Set the FCTN know to LD. Now enter the frequency”. The focus in COMET is on the various methods used to coordinate these media. This is achieved through bidirectional communication between the text and graphics generators.

Architecture

Figure 2.19 shows the system architecture of COMET. On receiving a request for an explanation, the content planner uses text plans, or schemas, to determine which information should be included from the underlying knowledge sources in the explanation. The content planner produces the full content for the explanation to be visualised before realisation takes place. The content is represented as a hierarchy of Logical Forms (LFs), which are passed to the media coordinator. The media coordinator refines the LFs by adding directives indicating which portions of the explanation are to be produced by each of the media generators. COMET includes text and graphics generators which both process the same LFs, producing fragments of text and graphics that are keyed to the LFs they instantiate. The media generators can also interact further with the media coordinator, allowing the generation of cross-references. The text and graphics output by the generators are then combined by the Media layout module, which will format the final presentation for the low-level rendering-and-typesetting software.

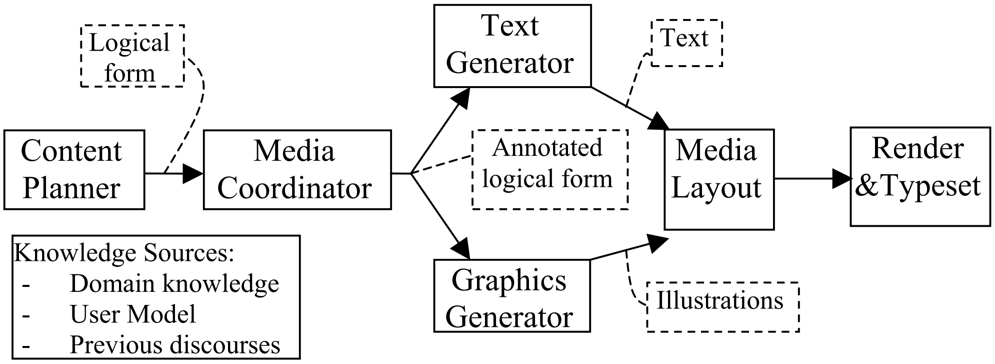
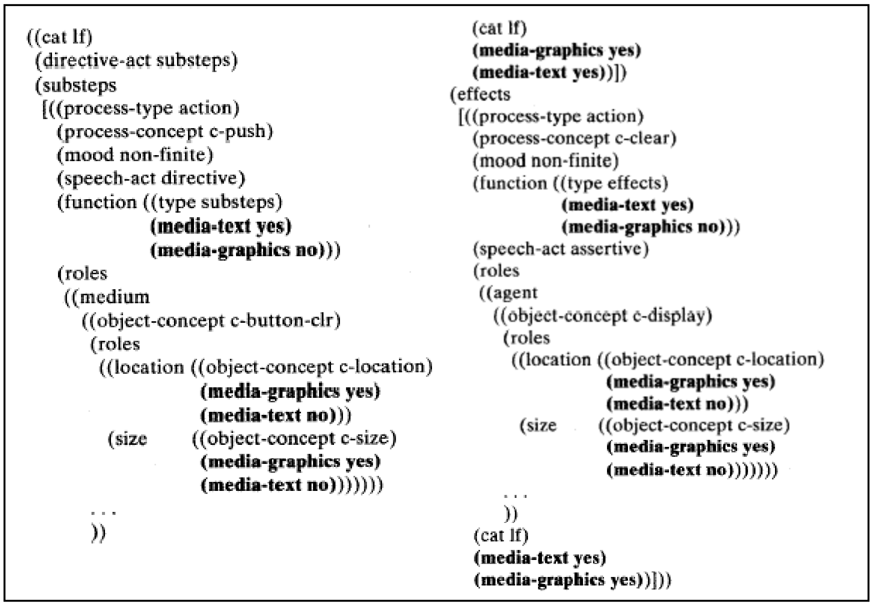


Figure 2.19 COMET system architecture

Semantic representation

As stated previously, the content of COMET is represented as a hierarchy of Logical Forms, which is a type of blackboard facility, a central repository in which a system component can record its intermediate decisions and examine those of other components. Each component reads and annotates its LFs, continually enriching it with further decisions and finer specifications until the explanation is complete. Annotations include directives (like the media coordinator’s choice of medium) and details about how a piece of information will be realised in text or graphics. While the annotated LF serves as a blueprint for the final explanation, it also allows for communication between media specific components. Using this technique COMET represents the semantics of an explanation. Figure 2.20 shows a portion of an annotated LF produced by the media coordinator in COMET. The majority of this LF was generated by the content planner, while the annotations added by the media coordinator are represented in bold. This LF specifies a single substep and its effect, where the substep is a simple action (c-push) (5th Line Of Code (LOC)) and the effect is a simple action (c-clear) (27th LOC). The c-push substep has one role (the medium, c-button-clr) (13th LOC), and it also specifies that the location and size of the object (button) should be included.



```
((cat lf)
 (directive-act substeps)
 (substeps
  [((process-type action)
    (process-concept c-push)
    (mood non-finite)
    (speech-act directive)
    (function ((type substeps)
               (media-text yes)
               (media-graphics no)))
    (roles
     ((medium
       ((object-concept c-button-clr)
        (roles
         ((location ((object-concept c-location)
                     (media-graphics yes)
                     (media-text no)))
          (size     ((object-concept c-size)
                     (media-graphics yes)
                     (media-text no))))))
        . . .
        ))
    (cat lf)
    (media-graphics yes)
    (media-text yes))])
  (effects
   [((process-type action)
    (process-concept c-clear)
    (mood non-finite)
    (function ((type effects)
               (media-text yes)
               (media-graphics no)))
    (speech-act assertive)
    (roles
     ((agent
       ((object-concept c-display)
        (roles
         ((location ((object-concept c-location)
                     (media-graphics yes)
                     (media-text no)))
          (size     ((object-concept c-size)
                     (media-graphics yes)
                     (media-text no))))))
        . . .
        ))
    (cat lf)
    (media-text yes)
    (media-graphics yes))]))
```

Figure 2.20: Example of Logical Form in COMET

Mapping to Output

In response to a user request for an explanation (e.g. "How do I load the transmission frequency into channel 1?") COMET (Feiner & McKeown 1991a,b) dynamically determines the explanation's content using constraints based on:

- the type of request
- the information available in the underlying knowledge base
- information about the user's background, discourse context, and goals

Having determined what to present, COMET decides how to present it at the time of generation with graphics and text. The pictures and text that it uses are not 'canned', i.e. it does not select from a database of conventionally authored text, pre-programmed graphics, or recorded video. Instead, COMET decides which information should be expressed in which medium, which words and syntactic structures best express the portion to be conveyed textually, and which graphical objects, style, and picture structure best express the portion to be conveyed graphically. Figure 2.21 shows an example text output from COMET in response to the user request: "How do I load the transmission frequency into channel 1?"

```
Set the channel knob to position 1.
Set the MODE know to SC.
Set the FCTN know to LD
Now enter the frequency:
        First, press the FREQ button.
        This will cause the display to show an arbitrary number.
        Next, press the CLR button in order to clear the display.
        Next, enter the new frequency using the number buttons.
        Next, record this number in order to check it later.
        Finally, press Sto ENT.
        This will cause the display to blink.
```

Figure 2.21: Text produced by COMET

Fusion, integration & coordination

COMET used a form of temporal reasoning to control representation and coordination. It determines which information is best expressed in text and which in graphics, and coordinates both these media using bidirectional interaction between its separate text and graphics generators. COMET focuses on several types of coordination to achieve this:

- coordination of sentence breaks with picture breaks to avoid splitting sentences over picture boundaries,
- the generation of cross references from text to graphics,
- and the influence of generation in one medium on the other.

Thus, the two main aspects to the media coordination in COMET are: (1) The use of a blackboard for common content-description using annotated LFs allows for more flexible interaction between media, making it possible for each generator to query and reference other generators. By passing the same annotated description of what is to be described to each generator permits each generator to use information about what the other generators present to influence its own presentation; (2) Bidirectional interaction between the media-specific generators is necessary for certain kinds of coordination. Bidirectional interaction allows COMET to generate explanations that are structurally coordinated and that contain cross-references between media. Therefore, the graphics generator can inform the language generator about the graphical actions it has decided to use, and the language generator can then produce coordinated text like "the highlighted knob in the left picture".

2.6.2 WIP

Similar to COMET, WIP (Wahlster et al. 1992) is another intelligent multimedia authoring system that presents mechanical instructions in graphics and language for assembling, using, maintaining, or repairing physical devices such as espresso machines, lawn mowers, or modems. The task of the knowledge-based presentation system in WIP is the generation of a variety of multimodal documents from input consisting of a formal description of the communicative intent of a planned presentation. WIP is a transportable interface based on processing schemes that are independent of any particular back-end system and thus requires only a limited effort to adapt to a new application or knowledge domains. Wahlster et al. focus on the generalisation of text-linguistic notions such as coherence, speech acts, anaphora, and rhetorical relations to multimedia presentations. For example, they slightly extend Rhetorical Structure Theory (Mann et al. 1992) to capture relations not only between text fragments but also picture elements, pictures, and sequences of text-picture combinations.

Architecture

The design of WIP follows a modular approach. WIP includes two parallel processing cascades for the incremental generation of text and graphics. In order to achieve a fine-grained and optimal division of work between the single system components, WIP allows for various forms of interaction between them. Besides interaction within the cascades, all components also have access to the design record that contains results generated so far. The major components of WIP as portrayed in Figure 2.22 are the presentation planner, the layout manager, and the generators for text and graphics.

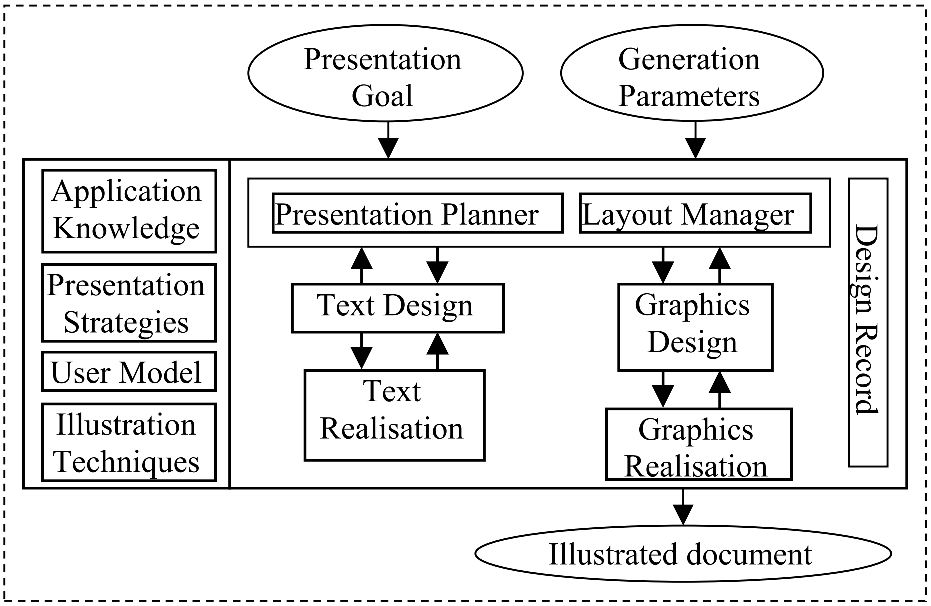


Figure 2.22 WIP system architecture

Semantic Representation

WIP uses an operator-based approach to planning that consists of 'planning strategies'. These strategies represent the presentation goal of the output using an approach to operationalise RST (Rhetorical Structure Theory). The strategies are represented by name, a header, an effect, a set of applicability conditions and a specification of main and subsidiary acts. Whereas the header of a strategy is a complex communicative act (e.g., to enable an action), its effect refers to an intentional goal (e.g., the user knows a particular object). Examples of presentation strategies are shown in Figure 2.23.

```
[S1] NAME:     Request-Enable-Motivate
      HEADER:  (Request P A ?action T)6
      Effect:      (BMB P A (Goal P (Done A ?action)))
     Applicability Conditions:
              (And
                (Goal P (Done A ?action))
                (Bel P (Complex-Operating-Action ?action))
                (Bel P (Agent ?agent ?action))
                (Bel P (Object ?object ?action)))
      Main Acts:
             (S-request P A (?action-spec (Agent ?agent-spec)
                                          (Object ?object-spec)))
             (Activate P A (Action ?action) ?action-spec T)
             (Activate P A (Agent ?agent) ? agent -spec T)
             (Activate P A (Object ?object) ?object-spec T)
      Subsidiary Acts:
             (Motivate P A ?action ?mode-1)
             (Enable P A ?action ?mode-2)
```

```
[S2] NAME:     Describe-Orientation
      HEADER:  (Describe P A (Orientation ?orientation)G)
      Effect:      (BMB P A (Has-Orientation ?orientation ?x))
     Applicability Conditions:
             (Bel P (Has-Orientation ?orientation ?x))
      Main Acts:
             (S-Depict P A (Orientation ?orientation) ?p-orientation ?pic)
      Subsidiary Acts:
             (Achieve P (BMB P A (Identifiable A ?x ?px ?pic)) ?mode)
```

```
[S3] NAME:     Provide-Background
      HEADER:  (Background P A ?x ?px ?picG)
      Effect:      (BMB P A (Identifiable A ?x ?px ?pic))
     Applicability Conditions:
                       (And
                         (Bel P (Image-of ?px ?x ?pic))
                         (Bel P (Perceptually-Accessible A ?x))
                         (Bel P (Part-of ?x ?z)))
      Main Acts:
                        (S-Depict P A (Object ?z) ?pz ?pic)
      Subsidiary Acts:
             (Achieve P (BMB P A (Identifiable A ?z ?pz ?pic)) ?mode)
```

Figure 2.23: WIP presentation strategy examples

The first strategy (S1) above can be used to request the user to perform an action. In this strategy, three kind of acts occur: the elementary act (BMB P A…), three referential acts for specifying the action (Bel P…) and the semantic case roles associated with the action (Activate P A…), and two complex communicative acts (Motivate P A… and Enable P A…). The second and third strategies may be employed to show the orientation of an object and to enable its identification in a picture.

Mapping to Output

When designing an illustrated document, an author has to decide which mode combination is the most suitable for meeting his goals; whereas, the automatic media generation process and decision-making process for mode selection is controlled by different factors including:

- target group/intended audience,
- target language,
- the kind of information content,
- the communicative functions that textual and pictorial parts ought to fill in a presentation,
- resource limitations (e.g. limitations due to the output medium, or space and time restrictions),
- user characteristics (e.g. trained or untrained in reading visual displays),
- the presentation objective/user's task.

WIP also incorporates media preferences for different information types. These specify:

- graphics over text for concrete information (e.g., shape, color, texture, also events and actions if visually perceptible changes)
- graphics over text for spatial information (e.g. location, orientation, composition) or physical actions and events (unless accuracy is preferred over speed, in which case text is preferred)
- text to express temporal overlap, temporal quantification (e.g. 'always'), temporal shifts (e.g. 'three days later') and spatial or temporal layout to encode sequence (e.g. temporal relations between states, events, actions)
- text to express semantic relations (e.g. cause/effect, action/result, problem/solution, condition, concession) to avoid ambiguity in picture sequences; graphics for rhetorical relations such as condition and concession only if accompanied by verbal comment
- text for quantification, especially most vs. some vs. exactly-n
- graphics to express negation (e.g. overlaid crossing bars) unless scope was ambiguous, then use text

Some of these preferences are captured in constraints associated with presentation actions, which are encoded in plan-operators, and use feedback from media realisers to influence the selection of content.

In the WIP system content and mode selection are interleaved using a uniform planning mechanism. This is possible since the presentation strategies and meta-rules accessed by the planner contain not only knowledge about what to present, but also knowledge about adequate mode combinations. In contrast to this, presentation planning and content realisation are performed by separate components that access disparate knowledge sources. This modularisation enables parallel processing, but makes interaction between the single components necessary. As soon as the planner has decided which generator should encode a certain piece of information, this piece is passed on to the respective generator. Conversely, the planning component immediately incorporates the results of the generators. Therefore, the processing of all components had to be 'interrupted' at certain points to allow other components to react. To cope with uncertainties concerning the results of other components, WIP's presentation planner maintains partial descriptions of unspecified variables through the use of constraints. Thus, it is able to continue planning without premature commitment.

Fusion, integration & coordination

In WIP text and graphics generators interact, for example, to generate unambiguous linguistic and visual references to objects. This interaction enables the text generator to make visual references such as 'The on/off switch is located in the upper left part of the picture'. WIP also includes a grid-based layout system that co-constrains the presentation planner.

More recent work of the authors focuses on interactive presentations, having an animated agent, called PPP persona (Personalised Plan-based Presenter) (André et al. 1996, André & Rist 2000), to navigate the presentation. PPP continues work done in WIP by adding three fundamental extensions:

- Planning multimedia presentation acts using of an animated character
- Interactive multimedia presentations allowing for user feedback
- Monitoring the effectiveness of a presentation by tracking users' behavior

Further work includes AiA (Adaptive Communication Assistant for Effective Infobahn Access) (André & Rist 2001) and Miau (Multiple Internet Agents for User-Adaptive Decision Support) (André et al. 2000). AiA develops a series of personalised information assistants that aim at facilitating user access to the Web, while Miau is investigating performances given by a team of characters as a new form of presentation.

2.6.3 TEXTPLAN

TEXPLAN (Textual EXplanation PLANner) (Maybury 1993, Maybury & Wahlster 1998) reasons about a hierarchy of communicative actions to accomplish particular discourse goals. It designs explanations in the form of narrated or animated route directions in the domain of a cartographic information system. It generates multimedia explanations, tailoring these explanations based on a set of hierarchically organised communicative acts with three levels: rhetorical, illocutionary (deep speech acts) and locutionary (surface speech acts). At each level these acts can be physical, linguistic or graphical. Physical acts include gestures (deictic), attentional or other forms of body language, linguistic acts are speech acts such as inform, request or command which characterise the illocutionary force of a single utterance, whilst graphical acts include highlighting, or zooming in/out, drawing and animating objects. TEXTPLAN defines several communicative acts, including those mentioned, physical, linguistic and graphical, in a common plan operator language.

Architecture

Figure 2.24 gives a sense of the hierarchy of communicative acts which is embodied in TEXTPLAN. TEXTPLAN plans higher level rhetorical actions such as; identify a given entity, compare two entities, or explain a process, in terms of more primitive illocutionary speech acts (e.g. inform, request) which in turn were further specified as locutionary or surface speech acts (e.g. assert, ask, command).

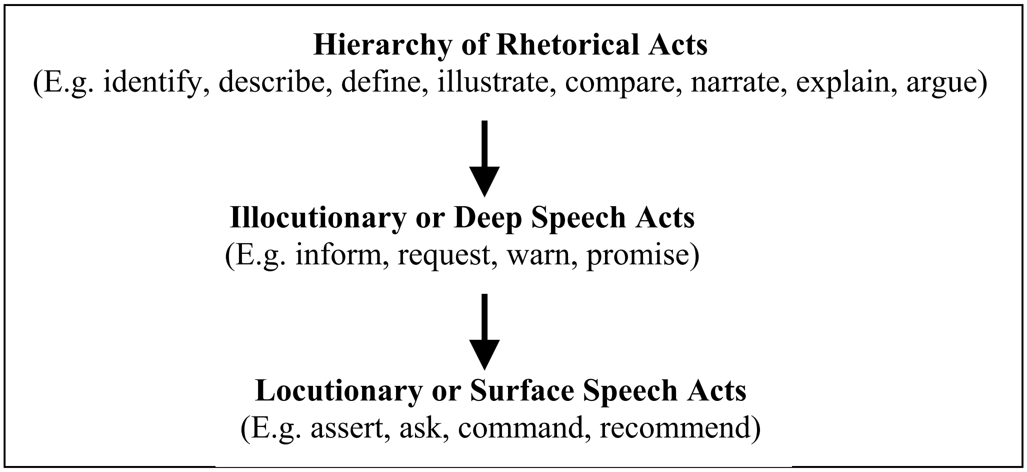

Hierarchy of Rhetorical Acts
(E.g. identify, describe, define, illustrate, compare, narrate, explain, argue)
Illocutionary or Deep Speech Acts
(E.g. inform, request, warn, promise)
Locutionary or Surface Speech Acts
(E.g. assert, ask, command, recommend)

Figure 2.24 Hierarchical Theory of Communicative Acts

A system designed to deliver explanations using each level of communicative act should be capable of explaining physically, linguistically or graphically, depending upon which type of explanation is best suited to the communicative goal.

Semantic representation

Similar to physical acts, communicative acts (rhetorical, linguistic, and graphical) can be formalised as plans. In TEXTPLAN communicative acts are represented as operators in the plan library of a hierarchical planner. Each plan operator defines the constraints and preconditions that must hold before a communicative act applies, it's intended effects, and the refinement or decomposition of the act into sub-acts. Plan operators are encoded in an extension of first order predicate calculus that allows for optionality within the decomposition. Figure 2.25 shows the *Identify-location-linguistically* plan operator, which is one of several methods of performing the communicative action- identify.

NAME	Identify-location-linguistically
HEADER	Identify(S,H, entity)
CONSTRAINTS	Entity?(entity)
PRECONDITIONS	WANT(S,KNOW(H, Location(entity)))
EFFECTS	KNOW(H, Location(entity))
DECOMPOSITION	Assert(S, H, Location(entity))

Figure 2.25: TEXTPLAN's Identify-location-linguistically plan operator

As defined in the HEADER of the plan operator, the *Identify* act takes three arguments, the speaker (S), the hearer (H), and an entity. The meaning of Figure 2.25 is as follows: Provided the third argument is indeed an entity-object/event (CONSTRAINTS) and the speaker wants the hearer to know about it (PRECONDITIONS), the speaker (S) will identify the location of the entity by informing the hearer (H) of its location (DECOMPOSITION), which has the intended effect that the hearer knows about it (EFFECTS).

Mapping to output

As previously stated, communicative acts in TEXPLAN can be arranged into three levels: rhetorical, illocutionary (deep speech acts) and locutionary (surface speech acts). At each level the system is designed to deliver explanations using combinations of physical, linguistic or graphical modalities. Which type of act to use for achieve a communicative goal is determined by a number of factors including: (1) the contents of the application knowledge base, (2) a model of the user's knowledge, beliefs, and plans, (3) a model of discourse, (4) the complexity of the actions (e.g. number of items in the decomposition).

Fusion, integration & coordination

If the object we are trying to identify using TEXPLAN has an associated graphical presentation, TEXTPLAN can augment natural language with visual identification within the explanation output. If the user asked TEXTPLAN "Where is Chemnitz?", the system could either give a unimodal text response- "Chemnitz is a town located at 50.82° latitude, 12.88° longitude", or a multimodal response including visual output along with the text output given above. The *Identify-location-linguistically-&-visually* plan operator in Figure 2.26 is selected only if its constraints are satisfied (i.e. the given entity is a cartographic entity e.g. town, road, lake), in which case the plan operator then ensures that the entity is visible. If the entity is out of the currently visible region or too small to be seen, this can be achieved by panning, jumping, or zooming to the region around the designated entity. In this way TEXTPLAN ensures that modalities are fused, integrated and coordinated correctly on output.

NAME	Identify-location-linguistically-&-visually
HEADER	Identify(S, H, entity)
CONSTRAINTS	Cartographic-Entity?(entity)
PRECONDITIONS	Visible(entity) ^ WANT(S,KNOW(H, Location(entity)))
EFFECTS	KNOW(H, Location(entity))

DECOMPOSITION	Indicate-Deictically(S, H, entity)
	Assert(S, H, Location(entity))

Figure 2.26: TEXTPLAN's Identify-location-linguistically-&-visually plan operator

2.6.4 CICERO

CICERO (Hovy & Arens 1993, Arens et al. 1993, Arens & Hovy 1995) is an application-independent multimedia interaction manager that performs run-time media coordination and allocation, so as to adapt dynamically to a presentation context. Its aim is to develop a reusable model of an intelligent manager that coordinates/integrates and synchronises the various media in a way that decreases system complexity caused by information overload. It is plug-in compatible with host information resources such as "briefing associate" workstations, expert systems, databases, etc., as well as with multiple media such as natural language, graphics, etc. An example text input to CICERO, when it is applied to the domain of Swiss cheese sales could be: "display the top three sales regions in 1988 and 1989". In response to this, as Figure 2.28 portrays, CICERO presents bar charts portraying the sales figures for these areas using information attained from the relevant databases.

Architecture

CICERO's architecture embodies a single high-level generic framework in which the information relevant to the various modules and data resources can be homogenised and integrated effectively in a single set of abstractions. Such a homogeneous framework not only enables a uniform interface between all modules, it allows for extensibility in that new modules (e.g. media, information sources) can be added with minimum overhead and become immediately useful to the system and the interface. Thus, CICERO is a generic interaction platform that can be tailored as required to be applied over and over again in new environments. Figure 2.27 shows the overall view of CICERO architecture and its major modules.

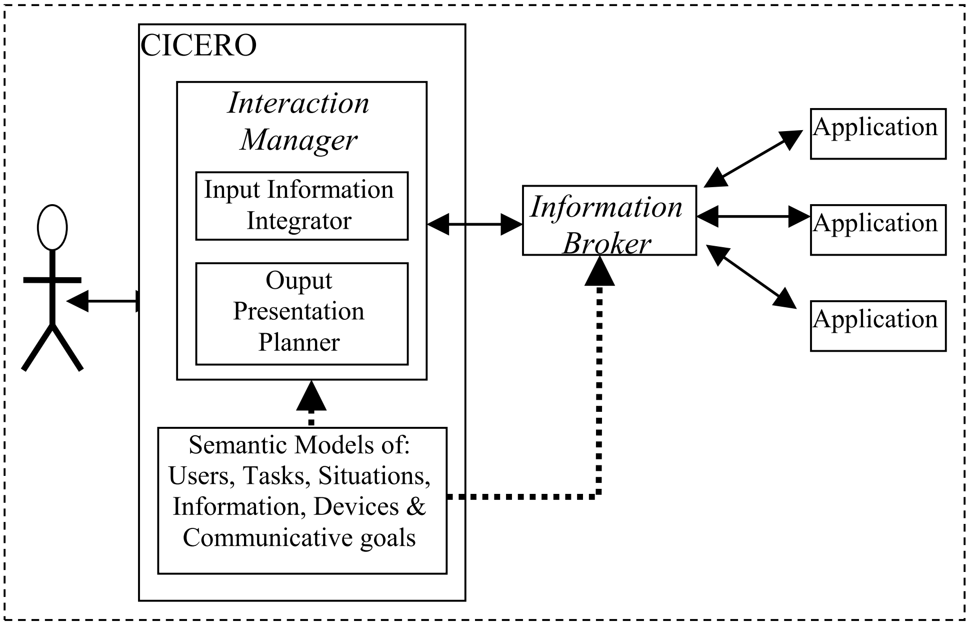

CICERO
Interaction Manager
Input Information Integrator
Ouput Presentation Planner
Semantic Models of: Users, Tasks, Situations, Information, Devices & Communicative goals
Information Broker
Application
Application
Application

Figure 2.27: CICERO architecture

The interaction manager (a) dynamically plans and coordinates the operation of the media and information resources at hand, (b) understands enough about the domain and the discourse to maintain an extended dialogue, and (c) supports plug-in compatibility, necessarily involves numerous components. Cicero's input facilities consist of modules that operate on and with a set of generic and specific semantic models. Three types of module are required to handle input: the input media themselves (with associated models); the information integrator module; and the input half of the discourse manager. The output presentation planner consists of two linked reactive planners, a discourse planner and a display planner, that perform runtime discourse construction and medium allocation planning respectively. The information broker is used only when more than one application is present; otherwise, the single application links directly to CICERO through the task and information models.

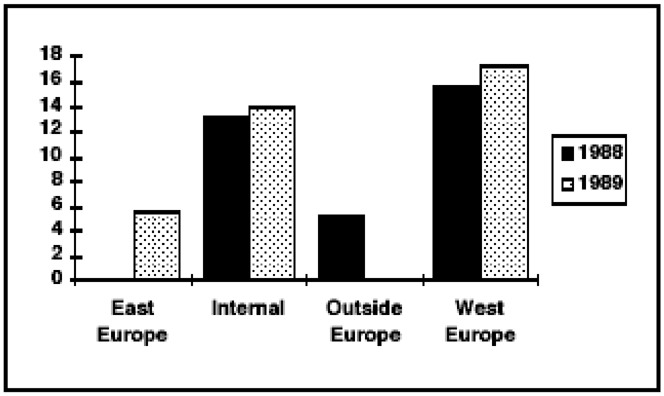


Figure 2.28: Example of CICERO output

CICERO is designed so that it can be reused in various contexts. To enable this type of presentation flexibility, the capabilities of each medium and the nature of the contents of each information source are semantically modeled as Virtual Devices and abstract information types respectively in a single uniform knowledge representation framework. These models facilitate extensibility by supporting the specification of new interaction behaviors and the inclusion of new media and information sources.

A Virtual Device is a model of a particular I/O functionality that may be realised by numerous different combinations of hardware and/or software. This model contains all relevant parameters defining the class of devices that perform the indicated function. Our preliminary studies indicate that each Virtual Device should contain, in addition to parameters for functionality, specifications and evaluation measures for (at least) the following:

- Hardware: limits of resolution; latency; device tolerances; etc.
- Software: data accuracy; data rate; etc.
- Human (cognitive): short-term memory requirements; communication protocol (language, icons, special terminology); etc.
- Human (physical): muscle groups and actions required; associated fatigue factors; etc.

- Usage: techniques of use (traditional menu vs. pie-chart menu, etc.); performance characteristics; etc.

Any actual device – that is, a concrete instantiation of the Virtual Device – should be measured using the standards and metrics specified in the Virtual Device parameter slots.

Semantic representation

The nature of the contents of each information source in CICERO are semantically modeled as abstract information types as mentioned previously. These information sources are the represented in the diagram of Cicero's architecture as semantic models. In CICERO there are abstract information types for modeling:

- characteristics of information to be displayed or input
- application task and interlocutors' goals
- discourse and communicative context
- user's goals, interests, abilities and preferences

The first of these information sources is concerned with the presentation-related characteristics of information. One of the methods used for this are 'internal semantic systems'. An example of such a semantic representation is given in Figure 2.29.

Carrier	*Internal Semantic System*
Picture	'real world' spatial location based on picture denotation
NL Sentence	'real world' sentence denotation
Table	Categorisation according to row and column
Graph	Coordinate values on graph axes
Map	'real world' spatial location based on map denotation
Ordered list	Ordinal sequenciality

Figure 2.29: Example of an internal semantic system

The information sources which are related to the application task and interlocutors' goals are semantically represented using an a partial classification of a producer's communicative goals. The discourse structure purpose and structure are done by operationalising and extending the relations of RST, which holds that textual discourses are coherent only if adjacent segments are related according to the constraints imposed by a small set of relations. The work on representing user's goals, interests, abilities and preferences in CICERO focuses on issues of human perception which influence the appropriateness of media choices for presentation of certain types of data.

Mapping to output

In other systems, information allocation is usually devolved to the system designer. The general challenge CICERO attempts to respond to is how to build a presentation-managing interface that designs itself at run-time so as to adapt to changing demands of information presentation. In order to achieve this in CICERO the system contains two linked reactive planners that perform run-time discourse construction and medium allocation planning, as outlined and tested in (Hovy & Arens 1993, Arens et al. 1993):

(1.) Content planning process (Planner 1): Operators construct the Discourse Structure from data elements and their interrelationships. Planning decisions are based upon:

- presenter's communicative goal(s),
- characteristics of data available for presentation,
- perceiver's knowledge, goals, interests, etc.

(2.) Media display planning process (Planner 2): Operators construct the Presentation Structure out of information in the Discourse Structure. Planning decisions are based upon:

- information currently displayed: data elements (from Discourse Structure),
- displayed configurations of data elements as larger identifiable units (from Presentation Structure),
- characteristics of media currently employed (on screen, etc.),
- characteristics of other media available at hand.

Based on these planning decisions CICERO can dynamically design its interface at run-time and hence, is in a much better position to respond to the ever-changing and hard-to-foresee display demands.

Fusion, integration & coordination

The core problem facing multimedia input in an interaction manager such as CICERO is the problem of integrating information from different media. For example, when a health care provider touches a specific field of a displayed table and types in the command "change this to 23.4 mg", the system must interpret the field touched as a particular data cell and then integrate the logical address of the cell with the deictic referent "this", in order to produce the user's complete disambiguated command. Deictic referents (such as "this" and "here") exist in language in order to allow the language user to use another, more suitable, medium, such as pointing. In addition to handling language deixes, deictic referents should be created also for the media of pointing and menus, as opportunities to allow the user to mix and match media in the most natural way. For both menus and pointing, this involves defining operations such as different button mouse clicks that will signal to the device drivers that additional information is about to be given about the currently selected item, using another medium.

For example, the health care provider may indicate the position of a tumor in the patient by clicking the left mouse button on a diagram of the torso. By clicking the right button instead, the user both indicates the tumor location and sets up a generic deictic reference, indicating that he or she will add something more about the tumor in another medium, such as typing. On encountering such a deictic referent, CICERO's information integrator must inspect the semantic types of all the most recent inputs from all media, and employ a set of generic and/or domain-specific concept matching heuristics to infer how to resolve the deixis. If, to continue the example, the health care provider also typed in "below the pancreas" after performing the right button click, CICERO's information integrator must then link up the locational information of the pointing gesture (defined by its nature to be of type *location* in the generic information model) with the English phrase (which is also interpreted as a location, since this is the definition of the preposition "below"), to create an enhanced output representation containing both pieces of information.

2.6.5 IMPROVISE

IMPROVISE (Illustrative Metaphor PROduction in VISual Environments) (Zhou & Feiner 1997, 1998) is an automated graphics generation system that can be used to design coordinated multimedia presentations. It is a knowledge based system that can automatically create sequences of animated illustrations to convey a wide variety of data in different application domains (e.g., computer network management application and health care application). IMPROVISE combines the two dominant approaches in graphics generation systems: the constructive approach and the parameterised one. Under the former approach, the representations are constructed from basic building blocks (visual variables) and then "glued" together to form larger units. The latter approach, however, uses visual models with parameters that are to be defined after the analysis of the data characteristics or attributes before the instantiation and interpretation of the contained information. The system is based on an extensible formalism to represent a visual lexicon, i.e. a collection of visual primitives, which can be accessed in the process of graphics generation. The term "primitive" in this case refers to a type of visual form (i.e. a "visual" word), which can be anything from a video clip to a 2D static text string. Parameterised visual forms are abstracted from this lexicon. The abstraction is general enough to cater for a range of types. The selection and instantiation of these abstract forms is subject to a number of syntactical, semantic, and pragmatic constraints.

Using two very different examples, IMPROVISE has demonstrated how it can create animated visual narratives in different situations. In the first of these IMPROVISE generates a visual narrative from scratch to present a hospital patient's information to a nurse after the patient's coronary artery bypass graft (CABG) operation. In this case, the generated visual narrative is combined with generated spoken sentences to produce a coordinated multimedia summary. In the second of these examples, IMPROVISE modifies an existing visual presentation of a computer network, using a set of animated visual actions to gradually reveal the internal structure of a user-selected network link.

Architecture

IMPROVISE's architecture is given in Figure 2.30. The system's presentation cycle starts with a set of presentation intents, which usually describe domain-specific communicative goals. The task analyser is responsible for formulating and translating presentation intents (e.g. to examine a network link structure) into corresponding visual tasks (e.g. focus on the link and expose its internals). To accomplish these visual tasks, the visual presentation planner starts the design process. In IMPROVISE, design is an interleaved process involving two submodules: the visual content planner and the visual designer.

IMPROVISE also has a simple interaction handler that processes user events and formulates new communicative goals (presentation intents), and these new goals are passed to the task analyser where a new design cycle begins. When embedded in a multimedia system, IMPROVISE has an additional communicative module, the messenger, which is responsible for exchanging information between other internal components of IMPROVISE and external components (e.g. a media coordinator, not shown here). IMPROVISE uses a task-based communicative strategy for its internal components to exchange messages with external components, and to let the messenger deal with all low-level format transformation (converting the data to a format that can be understood by another system) and communication issues (e.g. establishing socket connections).

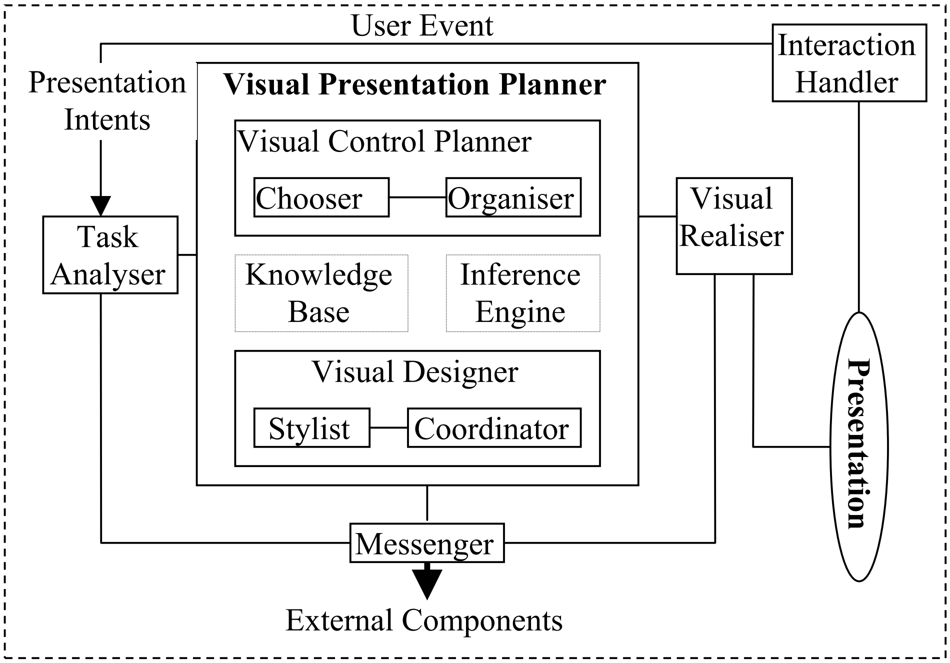


Figure 2.30 IMPROVISE architecture

Semantic representation

IMPROVISE is built to create a wide range of presentations for conveying heterogeneous data that can include both quantitative and qualitative information. As data semantics may vary greatly from one application to another, it is difficult to build a data model that captures all possible individual data features. To achieve this goal, IMPROVISE develops a knowledge-rich data semantic model based on a set of meta information which it uses to create graphical representations. The meta information is organised by a data characterisation taxonomy that abstracts a set of common visual presentation-related data properties. The taxonomy describes data characteristics along six dimensions:

type:	atomic vs. composite data entities
domain:	semantic categories, such as temperature or mass
attribute:	meta attributes, such as ordering, form and importance
relation:	data connections, such as has-part and attribute relations
role:	data function, such as being an identifier or a locator
sense:	visual interpretation preference, such as symbol or label

Not only do these meta properties help the system formulate portable (application-independent) design rules, but IMPROVISE can also use them to coordinate its actions with other media generators. Different media (e.g. graphics, speech) may also be used to convey different data entities or to present the same data entities from different perspectives. IMPROVISE fulfils this by creating a highlighting action:

Highlight <drips, ventilator, pacemaker>
(style OUTLINE) (startTime…) (endTime)

The action above specifies objects to be highlighted in a presentation, the highlighting style, and the associated temporal constraints. Similarly, a speech component produces a speech action:

Speak<cardionic-therapy>

Since both the highlighting and speech actions are associated with temporal constraints, they must be coordinated to produce a coherent presentation.

Mapping to output

IMPROVISE was designed as a knowledge-based system for automatically creating presentation for a wide variety of data. IMPROVISE can also stand-alone to create purely visual presentations or cooperate with other media generator to create multimedia presentations. To create a coherent multimedia presentation, IMPROVISE has a well formulated knowledge base, a sophisticated and efficient inference engine, and a portable visual realiser. The action-based inference engine employed by IMPROVISE allows for creation of coherent animated graphical illustrations. Given a set of inputs, including the data to be conveyed, the presentation intent, and the relevant presentation context, the engine can automatically create animated illustrations using sequences of temporally-ordered visual actions (instantiated visual techniques). The core of the inference engine is a top-down, hierarchical-decomposition, partial-order planner that employs visual techniques as planning operators and visual design principles as planning constraints. Using a top-down hierarchical-decomposition strategy, IMPROVISE first sketches a visual design that is complete, but too vague to be realised; then it refines the vague parts of the design into more detailed subdesigns, until the design had been refined to sequences of visual actions that can be executed by a realiser.

Fusion, integration & coordination

IMPROVISE can be used stand-alone to create purely visual presentations or cooperate with other media generators (e.g. a spoken language generator) to create coordinated multimedia presentations. There are four main features in IMPROVISE which play key roles in allowing for the design of coordinated multimedia presentations. The system's data representation formalism and temporal model of visual techniques help provide useful information to other media components in the course of designing a coordinated multimedia presentation. The action-based inference engine can cooperate with other media generators to produce desirable visual actions for a multimedia presentation and the visual realiser synchronises the execution of multiple media actions.

In this section a selection of intelligent multimedia presentation systems have been detailed. For each presentation system various elements were discussed including: architecture or system design, semantic representation(s) utilised by the system, output presentation (under what constraints or using what rules), and fusion, integration and coordination of modalities. The approaches used in these multimedia systems will inspire methods within TeleMorph's multimedia presentation planning.

2.7 Intelligent multimedia interfaces

Examples of intelligent multimedia dialogue systems include AIMI, AlFresco, CUBRICON and XTRA. Typically, these systems parse integrated input and generate coordinated output. Similar to the intelligent multimedia presentation systems discussed earlier there are approaches to media synchronisation and coordination in these multimedia interfaces which will be used to inspire TeleMorph's intelligent multimedia interface.

2.7.1 AIMI

AIMI (Burger & Marshall 1993) is an intelligent multimedia interface which can engage a user in a multimedia dialogue including natural language and graphical deixis in the context of a mission planning system. To fulfill this task the user is provided with maps, tables, charts and text, which are sensitive to further interaction through pointing gestures and other modalities. For example, a user might query in natural language "What aircraft are appropriate for the rescue mission?", to which the system might respond by automatically generating a table of appropriate aircraft. If the user then pointed to an item on the generated table of appropriate aircraft, this would be introduced into the discourse context so that she could simply ask "What is its speed?". Similarly, if a window containing a graphic was in focus in the dialogue, the user could anaphorically say "print that chart". AIMI uses non-speech audio to convey the speed and duration of processes which are not visible to the user.

2.7.2 AlFresco

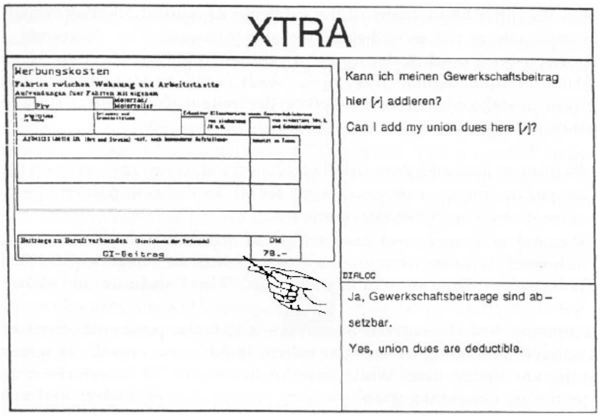
AlFresco (Stock et al. 1993) is an interactive, natural language centered system for presenting information about Fourteenth century Italian frescoes and monuments. It combines natural language processing with hypermedia to provide an efficient and user-oriented way to browse around through a pre-existing hyper-textual network. The combination of natural language and hypermedia gives rise to an advantage that while prefabricated hypermedia texts can compensate for a lack of coverage in the natural language modules, the on-demand generation of user-tailored texts can help to overcome the disorientation problem in the hypermedia environment. Also, further information about pictures and videos can be accessed by asking questions in natural language combined with direct pointing actions as in AIMI. With respect to user input, efforts in ALFresco have focussed on combining the interpretation of natural language deictic references with pointing to images displayed on a touch screen. For instance, the user can ask a question such as: ”Who is this person?” while touching the image of a person appearing in the currently displayed fresco. A dialog fragment taken from a session with ALFRESCO oriented to requests of information about Ambrogio Lorenzetti (A.L.) follows:

U: I would like to know if A.L. ever painted a work that represents a sacred scene with an angel.
S: Yes, for example, A Lorenzetti, the annunciation. The angel is the Archangel Gabriel.
U: Who is this person? [pointing at Mary on the touch screen]
S: Mary.
U: Can you show me an enlargement containing her?
S: [The system shows an enlargement]
U: Was the author influenced by some other painter?
S: Yes. For example by Giotto
U: Where did this master see the light? [meaning “where was he born”]
S: Colle di Vespignano.
U: Is there a work by A.L. in Trento?
S: No.
U: A fresco by Giotto in Assisi?
S: Yes. For example the sermon to the birds
U: Show me a detail with this scene.[pointing to some place about the center of the scene]
S: [The system shows the detail]
U: In what town was A.L. born?
S: Siena.
U: Show it to me.
S: [the system shows a small film segment about Siena]
U: I would like to see a fresco of the master in a monument of the town.
S: [the system shows a video segment about a large fresco, “the Effects of Good Government,” and outputs the name with a “description” button.]

Both the linguistic and the graphical deixes may be ambiguous, but usually the correct integration of the information provided by the two channels allows the identification of the intended reference. On output, images, videos and generated text are yielded that offer entry points for further hypertextual exploration. For instance, there is an underlying canned hypertext that includes critics' opinions about frescoes and painters, and descriptions of towns and monuments. The result is that the user communicates linguistically and by manipulating various entities, images, and the text itself.

2.7.3 XTRA

XTRA (eXpert TRAnslator) (Wahlster 1998) is an intelligent multimodal interface to expert systems that combines natural language, graphics, and pointing for input and output. XTRA provides natural language access to an expert system, which assists the user in filling out a tax form. During the dialog, the relevant page of the tax form is displayed on one window of the screen, so that the user can refer to regions of the form by tactile gestures. As shown in Figure 2.31, there are two other windows on the right part of the display, which contain the natural language input of the user (upper part) and the system's response (lower part). An important aspect of the communicative situation realised in XTRA is that the user and the system share a common visual field - the tax form. The user and system could both refer to regions in a tax form, without a pre-definition of pointing-sensitive areas. XTRA generates natural language and pointing gestures automatically, but relies on pre-stored tax forms. It represents not only the linguistic context, but also maintains a data structure for graphics to which the user and the system may refer during the dialogue. In the tax domain, the graphical context corresponds to a form hierarchy that contains the positions and the size of the individual fields as well as their geometrical and logical relationships.



XTRA

Werbungskosten

Fahrten zwischen Wohnung und Arbeitsstaette

Aufwendungen fuer Fahrten mit eigenem

Pkw

Motorrad/
Motorroller

Beitraege zu Berufsverbaenden

GI-Beitrag

DM

78.-

Kann ich meinen Gewerkschaftsbeitrag hier [↗] addieren?

Can I add my union dues here [↗]?

DIALOG

Ja, Gewerkschaftsbeitraege sind ab-setzbar.

Yes, union dues are deductible.

Figure 2.31: XTRA interface

2.7.4 CUBRICON

CUBRICON (Calspan-UB Research center Intelligent CONversationalist) (Neal and Shapiro 1991) is a system for Air Force Command and Control. The combination of visual, tactile and gestural communications is referred to as the unified view of language. CUBRICON produces relevant output using multimedia techniques. The user can, for example, ask, "Where is the Dresden airbase?", and CUBRICON would respond (with speech), "The map on the color graphics screen is being expanded to include the Dresden airbase." It would then say, "The Dresden airbase is located here," as the Dresden airbase icon and a pointing text box blink. Neal and Shapiro addressed the interpretation of speech and mouse/keyboard input by making use of an Augmented Transition Network grammar that uses natural language with gesture constituents. CUBRICON includes the ability to generate and recognise speech, to generate natural language text, to display graphics and to use gestures made with a pointing device. The system is able to combine all the inputs into the language parsing process and all the outputs in the language generation process. Figure 2.32 shows the architecture of CUBRICON.

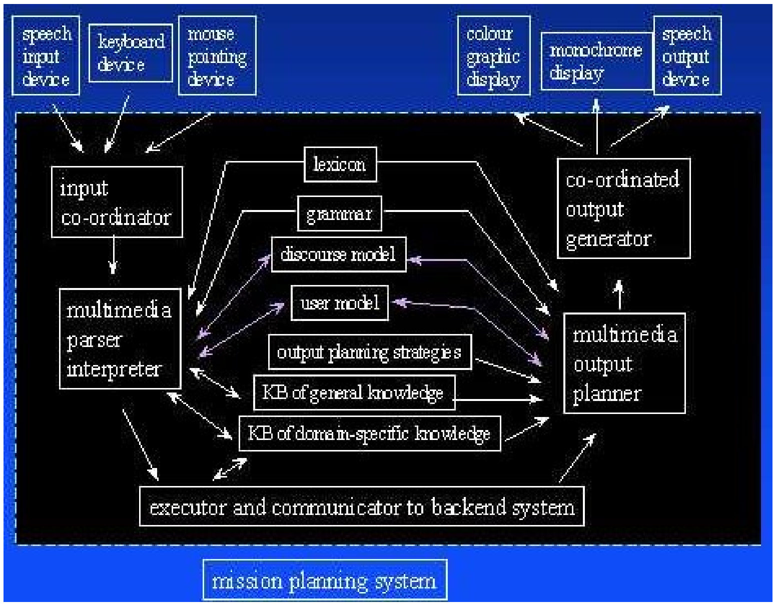


Figure 2.32: CUBRICON architecture

In this section examples of intelligent multimedia dialogue systems have been discussed including AIMI, AlFresco, XTRA and CUBRICON. Similar to the intelligent multimedia presentation systems discussed earlier there are approaches to media synchronisation and coordination in these multimedia interfaces which will be used to inspire TeleMorph's and TeleTuras' intelligent multimedia interface.

2.8 Intelligent multimedia agents

Embodied Conversational Agents are a type of multimodal interface where the modalities are the natural modalities of face-to-face communication among humans, i.e. speech, facial expressions, hand gestures, and body stance. Animated agents of this nature would prove very useful for TeleTuras' interface. Implementing an intelligent agent of this nature along with other modalities provides a much more natural, flexible and learnable interface for TeleTuras thus increasing the usability of the interface.

2.8.1 REA

Cassell et al. (2000) discuss REA (Real Estate Agent) (Figures 2.33a,b), which is an animated human simulation on a screen that can understand the conversational behaviors of the human standing in front of it via computer vision techniques, and responds with automatically generated speech and face, hand gesture and body animation. The system consists of a large projection screen on which REA is displayed and in front of which the user stands. Two cameras mounted on top of the projection screen track the user's head and hand positions. Users wear a microphone for capturing speech input. REA's application domain is real estate and it acts as a real estate salesperson which interacts with users to determine their needs, shows them around virtual properties, and attempts to sell them a house. Real estate sales was selected as an application area for REA since there are opportunities for both task-oriented and socially-oriented conversation. REA actually synthesises responses--including speech and accompanying hand gestures--based on a grammar and lexicon and communicative context.

Figures 2.33: User interaction with REA

2.8.2 BEAT

The system described above, REA, integrates a natural language generation engine (SPUD), and an animator's tool, BEAT (Cassell et al. 2001), which allows animators to input typed text that they wish to be spoken by an animated human figure. In the same way as Text-to-Speech (TTS) systems realise written text in spoken language (McTear 2002) BEAT realises written text in embodied expressive verbal and nonverbal behaviors such as face expression, head nods, gaze, and hand gestures. And in the same way as TTS systems are permeable to trained users, allowing them to tweak intonation, pause-length and other speech parameters, BEAT is permeable to animators, allowing them to write particular gestures, define new behaviours and tweak the features of movement. Figure 2.34 portrays an example of output from BEAT where it tells the user: "You just have to type in some text."

Figures 2.34: User interaction with BEAT

2.8.3 SAM

Sam (Cassell et al. 2000) (Figure 2.33), another 3D animated conversational agent, can tell stories and share experiences together with children by sharing physical objects across real and virtual worlds. It acts as a peer playmate in a shared collaborative space by using the real-time video of the child's environment as Sam’s background, so that Sam seems to exist in the child's play space. In this system the designers ensured that Sam, Sam's toy, and all aspects of the system and interaction were gender-neutral. Sam represents a five-year-old child who tells stories about its toy character in the magic castle. Audio recorded from an eight-year-old girl, which sounded only slightly feminine was used, but that was then offset with the somewhat masculine (although still androgynous) name of "Sam".

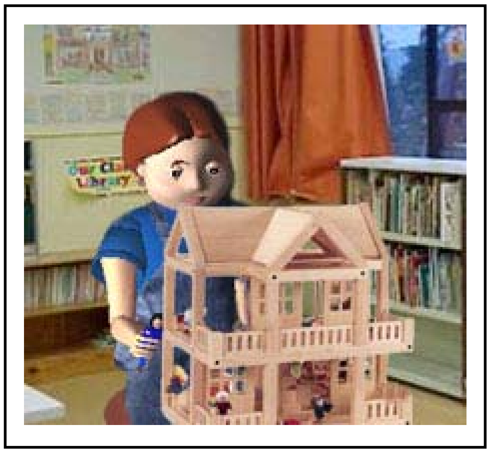


Figure 2.35: Example screenshot from SAM

2.8.4 Gandalf

Thórisson (1996) has built a system that addresses many issues in face-to-face communication. His agent, *Gandalf*, is rendered as a computer-animated face and associated hand. Gandalf is the interface of a blackboard architecture called *Ymir* which includes perceptual integration of multimodal events, distributed planning and decision making, layered input analysis and motor-control with human-like characteristics and an inherent knowledge of time. People interacting with the system must wear sensors and a close microphone to enable Gandalf to sense their position, sense what they are looking at and their hand position over time, and perform speech recognition. Figure 2.36 shows a user interacting with Gandalf. Using this system Thórisson has tested his theory for psychologically motivated, natural, multimodal communication using speech, eye contact, and gesture. Gandalf participates in conversations by attempting to produce all of these modalities at moments appropriate to the ongoing conversation. Because of the focus of Thórisson’s research, Gandalf does not attempt to express a personality, have realistic 3D graphic representation, other body movement (besides hand gestures) outside of the conversation, or other aspects needed for autonomous agents.

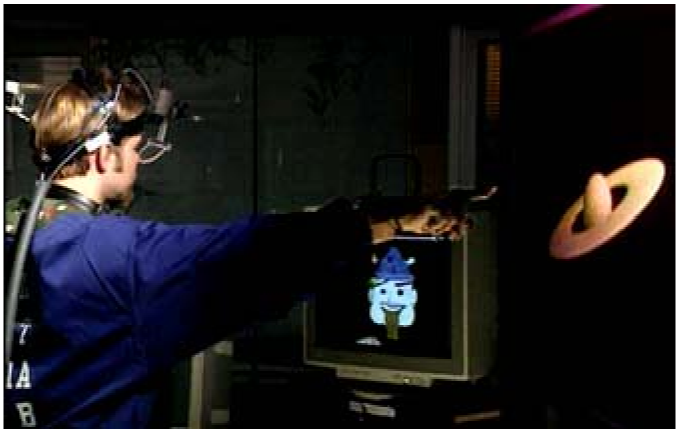


Figure 2.36: User interacting with Gandalf

Gandalf also uses canned text rather than performing natural language generation in answering. Nevertheless, the techniques used in Gandalf address a subset of the requirements for language use in agents, and could clearly be useful in multimodal communication with agents.

In this section Embodied Conversational Agents as a type of multimodal interface have been discussed. REA, BEAT, Sam and Gandalf have all been described. Details of these conversational agents have been given in terms of how they simulate natural human input and output modalities. The systems' specific domains (where they serve as conversationalist agents presenting information and explanations to the user) were also described. Along with system descriptions, examples of users interacting with each conversationalist agent were illustrated. Similar to the intelligent multimedia presentation systems and intelligent multimedia interfaces discussed earlier there are approaches to media synchronisation and coordination in these intelligent multimedia agents that will be used to inspire TeleMorph's and TeleTuras' design.

2.9 Cognitive Load theory (CLT)

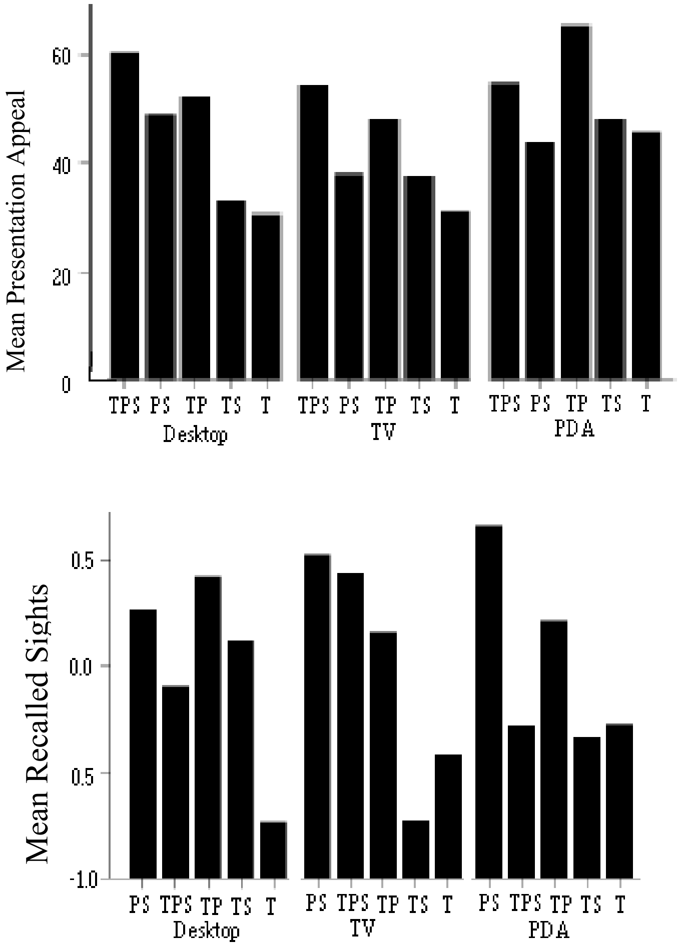
Elting et al. (2001) explain the cognitive load theory where two separate sub-systems for visual and auditory memory work relatively independently. The load can be reduced when both sub-systems are active, compared to processing all information in a single sub-system. Due to this reduced load, more resources are available for processing the information in more depth and thus for storing in long-term memory. This theory however only holds when the information presented in different modalities is not redundant, otherwise the result is an increased cognitive load. If however multiple modalities are used, more memory traces should be available (e.g. memory traces for the information presented auditorially and visually) even though the information is redundant, thus counteracting the effect of the higher cognitive load.

Elting et al. investigated the effects of display size, device type and style of Multimodal presentation on working memory load, effectiveness for human information processing and user acceptance. The aim of this research was to discover how different physical output devices affect the user's way of working with a presentation system, and to derive presentation rules from this that adapt the output to the devices the user is currently interacting with. They intended to apply the results attained from the study in the EMBASSI project where a large set of output devices and system goals have to be dealt with by the presentation planner.

Accordingly, they used a desktop PC, TV set with remote control and a PDA as presentation devices, and investigated the impact the multimodal output of each of the devices had on the users. As a gauge, they used the recall performance of the users on each device. The output modality combination for the three devices consisted of:

- plain graphical text output (T),
- text output with synthetic speech output of the same text (TS),
- a picture together with speech output (PS),
- graphical text output with a picture of the attraction (TP),
- graphical text, synthetic speech output, and a picture in combination (TPS).

The results of their testing on PDAs are relevant to any mobile multimodal presentation system that aims to adapt the presentation to the cognitive requirements of the device. Figure 2.37A shows the presentation appeal of various output modality combinations on various devices and Figure 2.37B shows mean recall performance of various output modality combination outputs on various devices.



(A) Mean Presentation Appeal

(B) Mean Recall Performance

Figure 2.37 Shows most effective and most acceptable modality combinations

The results show that in the TV and PDA group the PS combination proved to be the most efficient (in terms of recall) and second most efficient for desktop PC. So pictures plus speech appear to be a very convenient way to convey information to the user on all three devices. This result is theoretically supported by Baddeley's "Cognitive Load Theory" (Baddeley & Logie 1999, Sweller et al. 1998), which states that PS is a very efficient way to convey information by virtue of the fact that the information is processed both auditorally and visually but with a moderate cognitive load. Another phenomenon that was observed was that the decrease of recall performance in time was especially significant in the PDA group. This can be explained by the fact that the work on a small PDA display resulted in a high cognitive load. Due to this load, recall performance decreased significantly over time. With respect to presentation appeal, it was not the most efficient modality combination that proved to be the most appealing (PS) but a combination involving a rather high cognitive load, namely TPS (Fig. 2.37). The study showed that cognitive overload is a serious issue in user interface design, especially on small mobile devices.

From their testing Elting et al. discovered that when a system wants to present data to the user that is important to be remembered (e.g. a city tour) the most effective presentation mode should be used (Picture & Speech) which does not cognitively overload the user. When the system simply has to inform the user (e.g. about an interesting sight nearby) the most appealing/accepted presentation mode should be used (Picture, Text & Speech). These points should be incorporated into multimodal presentation systems to achieve ultimate usability. This theory will be used in TeleMorph in the decision making process which determines what combinations of modalities are best suited to the current situation when designing the output presentation, i.e. whether the system is presenting information which is important to be remembered (e.g. directions) or which is just informative (e.g. information on a tourist site).

2.10 Causal Probabilistic Networks (CPNs)

Bayesian networks are also called Bayes nets, Causal Probabilistic Networks (CPNs), Bayesian Belief Networks (BBNs) or simply belief networks (Jensen & Jianming 1995). A Bayesian Network (BN) is used to model a domain containing uncertainty in some manner. It consists of a set of nodes and a set of directed edges between these nodes. A Belief Network is a Directed Acyclic Graph (DAG) where each node represents a random variable. Each node contains the states of the random variable it represents and a conditional probability table (CPT) or in more general terms a conditional probability function (CPF). The CPT of a node contains probabilities of the node being in a specific state given the states of its parents. Edges represent probabilistic dependencies between these variables (nodes), i.e. cause-effect relations within the domain. These effects are normally not completely deterministic (e.g. disease -> symptom). The strength of an effect is modelled as a probability. The following example gives the conditional probability of having a certain medical condition given the variable temp (where P(…) represents a probability function):

- (1) If tonsillitis then P(temp>37.9) = 0.75
- (2) If whooping cough then P(temp>37.9) = 0.65

One could be lead to read 1) and 2) as rules. They shouldn't be. So a different notation is used (where | represented a directed edge in the bayesian network from the latter node (whooping cough) to the first node (temp>37.9)):

- P(temp>37.9 | whooping cough) = 0.65

If 1) and 2) are read as 'If otherwise healthy and...then...', there also needs to be a specification of how the two causes combine. That is, one needs the probability of having a fever if both symptoms are present and if the patient is completely healthy. All in all one has to specify the conditional probabilities:

- P(temp>37.9 | whooping cough, tonsillitis)

Where 'whooping cough' and 'tonsillitis' each can take the states yes and no. So, one must for any node specify the strength of all combinations of states for the possible causes. Inference in a Bayesian network means computing the conditional probability for some variables given information (evidence) on other variables. This is easy when all available evidence is on variables that are ancestors (parent nodes) of the variable(s) of interest. But when evidence is available on a descendant of the variable(s) of interest, one has to perform inference against the direction of the edges. To this end, Bayes' Theorem is employed:

- $P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$

In other words, the probability of some event A occurring given that event B has occurred is equal to the probability of event B occurring given that event A has occurred, multiplied by the probability of event A occurring and divided by the probability of event B occurring. Contradictory to the methods of rule based systems, the updating method of Bayesian networks uses a global perspective, and if model and information are correct, it can be proved that the method calculates the new probabilities correctly (correctly regarding the axioms of the classical probability theory). Any node in the network can receive information as the method doesn't distinguish between inference in or opposite to the direction of the edges. Also, simultaneous input of information into several nodes will not affect the updating algorithm. An essential difference between rule based systems and systems based on Bayesian networks is that in rule based systems you try to model the experts way of reasoning (hence the name expert systems), whereas with Bayesian networks one tries to model dependencies in the domain itself. The latter are often called decision support systems or normative expert systems.

Causal Probabilistic Networks are a technique for modeling the state of a domain, incomplete knowledge of the state of the domain at the time where a given task is to be performed, randomness in the mechanisms governing the behaviour of the domain, or a combination of these. CPNs are relevant to TeleMorph as it represents a technique for modeling information within the presentation design module of the TeleMorph architecture.

3. Project proposal

In this chapter a mobile intelligent multimedia presentation architecture called – TeleMorph for dynamically generating intelligent multimedia presentations is proposed. The multimodal output presentation which TeleMorph designs incorporates modalities that are determined by constraints that exist on a mobile device's wireless connection, the mobile device itself and also those limitations imposed by and experienced by the end user of the device. The unique contribution in TeleMorph is that the output presentation will consist of modalities that are determined primarily by the available bandwidth on the mobile network connecting the device. As bandwidth fluctuates continuously there is a necessity to constantly monitor this and adapt the multimedia presentation output modalities as necessary. Also, TeleMorph will consider the other constraints that affect an output presentation destined for a mobile device and will prioritise these based on the consequences of their depreciation or improvement. TeleMorph will use the Causal Probabilistic Networks (CPN) approach to reasoning and decision making in order to analyse a union of all relevant constraints imposed on TeleMorph and decide on the optimal multimodal output presentation, thus resulting in a comprehensive constraint-awareness system. TeleTuras, a tourist information aid for the city of Derry, will be developed as the testbed application of TeleMorph. TeleTuras will communicate TeleMorph-adapted presentations to tourists, focusing on the output modalities used to communicate information and also the effectiveness of this communication.

3.1 Architecture of TeleMorph

The architecture of TeleMorph is depicted in Figure 3.1. TeleMorph consists of server and client components. The TeleMorph server is concerned mainly with processing input from the client and producing an output presentation in response to the user's query, whilst the TeleMorph client is concerned with receiving the streamed media presentation from the server and displaying it in the *Media Player*.

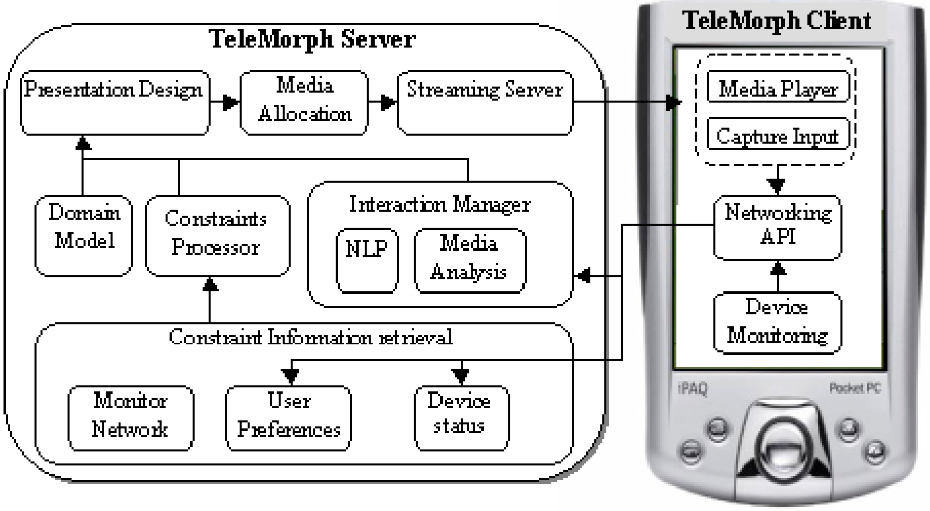


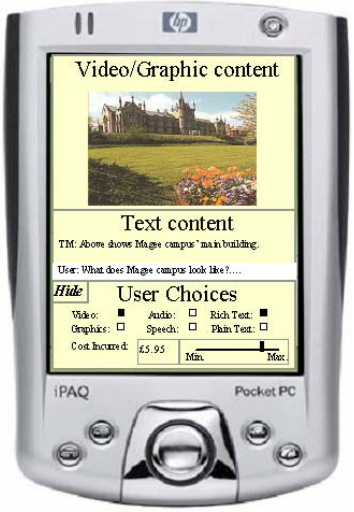
Figure 3.1: Architecture of TeleMorph

3.2 TeleMorph client

The TeleMorph client consists of a *Media Player* running on the client device in conjunction with the client *Capture Input* module that controls the input modalities consisting of textual, haptic deixis and speech input. The *Device Monitoring* will monitor client devices for fluctuations in memory availability, processing power, output abilities. The *Networking API* module will communicate with the server side using sockets.

3.2.1 Client output

The *Media Player* receives presentations from the *Streaming Server* and displays the multimedia content in specific areas of the display. As can be seen from Figure 3.2 the display contains panes for Video, Graphics and Text output. Output may also consist of Audio in the form of synthesised speech/Text To Speech (TTS), non-speech audio (music) and animation if an autonomous agent is utilised.



TM: Above shows Magee campus' main building.

User: What does Magee campus look like?....

Video: ■ Audio: □ Rich Text: ■
Graphics: □ Speech: □ Plain Text: □
Cost Incurred: £5.95 Min. Max.

Figure 3.2 Display on TeleMorph client

As TeleMorph will enable the end-user to determine the output modalities manually, a pane (which can be hidden) will be available which allows the user to select their modality preferences. A cost analysis sub-module will also be integrated into the *Capture Input* module which will calculate the cost incurred in downloading a presentation for the mobile customer. This will allow the user to directly manipulate the output modality combination (and each modality's quality/ resolution) in order to affect the total cost of downloading the media associated with the presentation. Fusion, integration and coordination of modalities deals with how a presentation should be arranged, in space and in time. The problem of temporal and spatial coordination across media will be addressed directly by elements within the semantic representation in the *Presentation Design* module in TeleMorph including: layout, structure, timing and synchronisation, and time manipulation elements.

3.2.2 Client input

Input to the TeleMorph Client application as shown in Figure 3.1 will be controlled by the *Capture Input* module, in the form of speech, text and haptic deixis input. Speech recognition will be controlled by a speech API that will be linked to the main client application. Text input is entered into the editable text box contained at the bottom of the Text content pane. Haptic deixis input will be controlled using graphics APIs. Input information will be sent back to the *Interaction Manager* on the server end of TeleMorph and analysed by the *Media Analysis* module. In this module input data (Language, Haptic) is analysed and various media types (speech/ text) are distinguished from one-another and processed appropriately. The speech recogniser which is positioned on the server will process the speech data received from the speech API component on the client. Speech recognition is a computationally heavy process so it is not feasible on a thin client. The *NLP* (Natural Language Processing) module processes the understanding of the user's language input to the system which will either be speech or text. Following this, all the input data is fused and their semantics interpreted.

3.3 TeleMorph server

The TeleMorph server consists of an *Interaction Manager* module as described above, a *Constraint Information Retrieval* module which obtains constraint data including networking, device and user constraints and also a *Constraint Processor* which will calculate the combined effect of these constraints. The latter will also consider Cognitive Load Theory (CLT) when evaluating the effect imposed by the combination of constraints. The *Interaction Manager* and *Constraints Processor* provide input for the *Presentation Design* module. A *Domain Model* is also incorporated into TeleMorph in order to specify the domain within which the system is being applied. The default domain model in TeleMorph will be tourism, as TeleTuras, the testbed for TeleMorph, will utilise this information. This model will contain specific knowledge about the city of Derry and a variety of relevant information. The *Presentation Design* module designs a presentation which consists of modalities determined by the fore-mentioned constraints and the domain model. Following this, the presentation media are allocated and fused by the *Media Allocation* module and then streamed to the mobile client by the *Streaming Server* to be presented to the user in the *Media Player*.

Constraints that TeleMorph considers when mapping the semantic representation to the output presentation include:

- Network: Bandwidth (also latency and Bit error rate)
- Device: display, available output modalities, memory, CPU
- User: preferences, cost incurred
- Cognitive Load Theory

Using information attained about these constraints the presentation design module will determine the most appropriate output modality combination for the output presentation. This will involve a rule based decision making process, which will need to be scalable in order to allow for the consideration of additional constraints. The methodology for implementing this is undecided as yet, but it is expected that an Artificial Intelligence

(AI) technique will be employed for this purpose. As this process will require the ability to reason, probabilistic reasoning could be employed. Causal Probabilistic Networks (CPNs) and Bayesian Belief Networks (BBNs) are modelling techniques based on probability theory may serve for this purpose.

In order to implement the TeleMorph architecture for initial testing, a scenario will be set up where switches in code will emulate fluctuating constraints. To implement this TeleMorph will use a table of constraint values ranging from ideal conditions to inferior conditions (e.g. bandwidths ranging from those available in 2G, 2.5G/GPRS and 3G networks). This table will also provide data representing the other constraints; device display abilities, processing power and memory; user modality preferences and cost incurred; and CLT. Each constraint value will have access to related information on the modality/combinations of modalities that can be streamed efficiently in that situation.

The modalities (video, speech, animation, graphics, text) available for each of the fore-mentioned constraint conditions will be determined by calculating the qualifying state each considered element (bandwidth, device memory etc.) should be in to stream the modality. Also the amalgamations of modalities that are feasible will be computed. This will provide an effective method for testing initial prototypes of TeleMorph. It is anticipated that CPNs or BBNs will eventually replace this process and provide a more scalable and efficient system.

The GPS aspect of TeleMorph has not been investigated fully yet. Although a location based service could potentially provide very useful information to a mobile device user, this focus of TeleMorph initially will be on the effective design of multimodal presentations.

3.4 Data flow of TeleMorph

The data flow within TeleMorph is shown in Figure 3.3 which details the data exchange among the main components. Figure 3.3 shows the flow of control in TeleMorph. The *Networking API* sends all input from the client device to the TeleMorph server. Each time this occurs, the *Device Monitoring* module will retrieve information on the client device's status and this information is also sent to the server.

On input the user can make a multimodal query to the system to stream a new presentation which will consist of media pertaining to their specific query. TeleMorph will receive requests in the *Interaction Manager* and will process requests via the *Media Analysis* module which will pass semantically useful data to the *Constraint Processor* where modalities suited to the current network bandwidth (and other constraints) will be chosen to represent the information. The presentation is then designed using these modalities by the *Presentation Design* module. The media are processed by the *Media Allocation* module and following this the complete multimodal SMIL presentation is passed to the *Streaming Server* to be streamed to the client device.

A user can also input particular modality/cost choices on the TeleMorph client. In this way the user can morph the current presentation they are receiving to a presentation consisting of specific modalities which may be better suited their current situation (driving/walking) or environment (work/class/pub). This path through TeleMorph is identified by the dotted line in Figure 3.3. Instead of analysing and interpreting the media, TeleMorph simply stores these choices using the *User Prefs* module and then redesigns the presentation as normal using the *Presentation Design* module.

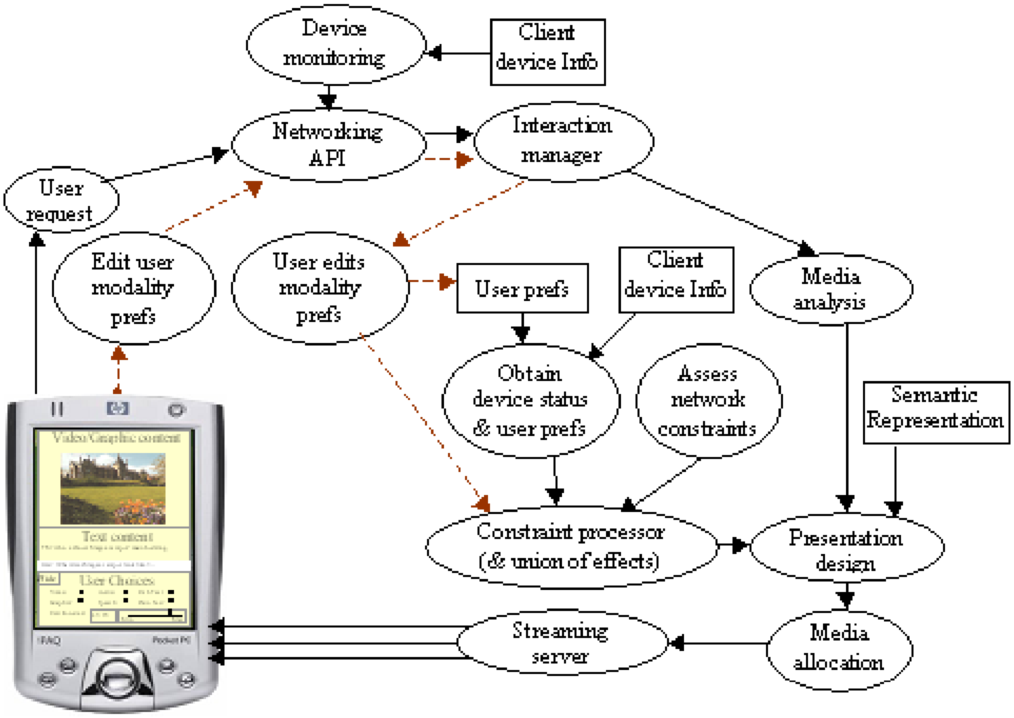


Figure 3.3 TeleMorph flow of control

The *Media Analysis* module that passes semantically useful data to the *Constraint Processor* consists of lower level elements that are portrayed in Figure 3.4. As can be seen, the input from the user is processed by the *Media Analysis* module, identifying Speech, Text and Haptic modalities. The speech needs to be processed initially by the speech recogniser and then interpreted by the *NLP* module. Text also needs to be processed by the *NLP* module in order to attain its semantics. Then the *Presentation Design* module takes these input modalities and interprets their meaning as a whole and designs an output presentation using the semantic representation. This is then processed by the *Media Allocation* modules as discussed previously.

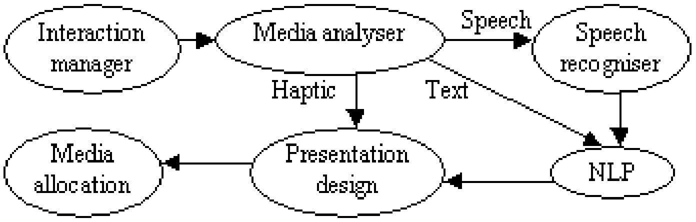


Figure 3.4 Media Analysis data flow

3.5 TeleTuras application

The domain chosen as a testbed for TeleMorph is *e*Tourism. The system to be developed called *TeleTuras* is an interactive tourist information aid for tourists in the city of Derry. It will communicate TeleMorph-adapted presentations to tourists consisting of: route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. The main focus will be on the output modalities used to communicate this information and also the effectiveness of this communication.

TeleTuras will be capable of taking input queries in a variety of modalities whether they are combined or used individually. Queries can also be directly related to the user's position and movement direction enabling questions/commands such as –

- "Where is the Millenium forum?"
- "Take me to the GuildHall"
- "What buildings are of interest in this area?"(Whilst circling a certain portion of the map on the mobile device, or perhaps if the user wants information on buildings of interest in their current location they need not identify a specific part of the map as the system will wait until the timing threshold is passed and then presume no more input modalities relating to this inquiry.).
- "Is there a Chinese restaurant in this area?"

The combined TeleMorph and TeleTuras system will entail the following processes:

- Receive and interpret questions from the user.
- Map questions to multimodal semantic representation.
- Match multimodal representation to database to retrieve answer.
- Map answers to multimodal semantic representation.
- Monitor user preference or client side choice variations.
- Query bandwidth status.
- Detect client device constraints and limitations.
- Generate multimodal presentation based on constraint data.

Applying a collection of common questions will test TeleTuras. These questions will be accumulated by asking prospective users/tourists what they would require from a tourist information aid such as TeleTuras.

3.6 Comparison with previous work

In the following tables there are comparisons showing features of various mobile intelligent multimedia (Table 3.1) and intelligent multimedia systems (Table 3.2). In the next two sections we will discuss the systems mentioning their features and shortcomings.

3.6.1 Mobile intelligent multimedia systems

Malaka et al. (2000, p. 22) points out when discussing DEEP MAP that in dealing with handheld devices "Resources such as power or networking bandwidth may be limited depending on time and location". From Table 3.1 it is clear that there are a wide variety of mobile devices being used in mobile intelligent multimedia systems. The issue of device diversity is considered by a number of the systems detailed in the Table. Some of these systems are simply aware of the type of output device (e.g. PDA, desktop, laptop, TV) (e.g. EMBASSI) and others are concerned with the core resources that are available on the client device (e.g. memory, CPU, output capabilities) (e.g. SmartKom-mobile). Some of these systems also allow for some method of user choice/preference when choosing output modalities in order to present a more acceptable output presentation for the end user. Pedersen & Larsen (2003) describe a test system which analyses the effect of user acceptance when output modalities are changed automatically or are changed manually by the user. This work is represented in Table 3.1 but as no final system was developed as part of the project. One other factor that is relevant to mobile intelligent multimedia systems is Cognitive Load Theory (CLT). This theory states the most efficient (judged by user retention) and the most appealing (user-acceptable) modalities for portraying information on various types of devices (PDA, TV, Desktop). One system that takes this theory into account is EMBASSI (Hildebrand 2000). One main issue which the systems reviewed fail to consider is the effect imposed by the union of all the aforementioned constraints. Of the mobile intelligent multimedia systems in Table 3.1, some acknowledge that (1) network bandwidth and (2) device constraints are important issues, but most do not proceed to take these into consideration when mapping their semantic representation to an output presentation, as can be seen from the table. As can also be seen from Table 3.1 none of the currently available mobile intelligent multimedia systems design their output presentation relative to the amount of available bandwidth available on the wireless network connecting the device. As discussed in section 2.2.2, the RAJA framework which is aimed at the development of resource-adaptive multi-agent systems was intended to provide resource-aware modules for DEEP MAP but this has not been integrated yet.

TeleMorph will differ with these systems in that it will be aware of all the constraints which have been mentioned. Primarily, TeleMorph will be bandwidth aware in that it will constantly monitor the network for fluctuations in the amount of data that can be transmitted per second (measured in bits per second (bps)). As mobile enabled devices vary greatly in their range of capabilities (CPU, memory available, battery power, input modes, screen resolution and colour etc), TeleMorph must also be aware of the constraints that exist on TeleMorph's client device and take these into consideration when mapping to output presentation. TeleMorph will also be aware of user-imposed limitations, which will consist of the user's preferred modalities and a restriction set by them on the cost they will incur in downloading the presentation. One other factor that will be considered when designing the output in TeleMorph is Cognitive Load Theory (CLT). TeleMorph will use CLT to assist in setting the output modalities for different types of information that may be portrayed in a presentation, such information which requires high levels of retention (e.g. a city tour), or information which calls for user-acceptance (purely informative) oriented modalities (e.g. information about an interesting sight nearby). From Figure 3.1 one can also identify that the combination of all these constraints as a union is also a unique approach. TeleMorph will be aware of all the relevant constraints that a mobile multimodal presentation system should be concerned with. TeleMorph will then analyse a union of these constraints and decide on the optimal multimodal output presentation. The method employed by TeleMorph to process these various constraints and utilise this information effectively to design the most suitable combinations of output modalities is the challenge that will prove core to this research.

Systems	**Device**	**Location Aware**	**Device Aware**	**User Aware**	**Cognitive Load Aware**	**Bandwidth Aware**	**Constraint union**
SmartKom-mobile	Compaq iPaq	●	●	●			
DEEP MAP	Xybernaut MA IV	●	●	●			
CRUMPET	Unspecified Mobile Device	●		●			
VoiceLog	Fujitsu Stylistic 1200 pen PC		●	●			
MUST	Compaq iPaq	●					
Aalborg (Koch 2000)	Palm V	●					
GuideShoes	CutBrain CPU	●					
The Guide	Mobile Phone	●		●			
QuickSet	Fujitsu Stylistic 1000	●	●				
EMBASSI	Consumer devices (e.g. Navigation System)		●		●		
Pedersen & Larsen (2003)	Compaq iPaq	●	●	●			
TeleMorph	J2ME device		●	●	●	●	●

Table 3.1 Comparison of Mobile Intelligent Multimedia Systems

3.6.2 Intelligent multimedia systems

Table 3.2 shows that TeleMorph & TeleTuras utilise similar input and output modalities to those employed by other mobile intelligent multimedia presentation systems. One point to note about the intelligent multimedia presentation systems in Table 3.2 is that on input none of them integrate vision, whilst only one system uses speech and two use haptic deixis. In comparison, all of the mobile intelligent multimedia systems in the Table integrate speech and haptic deixis on input. Both Guide and Quickset use only text and static graphics as their output modalities, choosing to exclude speech and animation modalities. VoiceLog is an example of one of the mobile systems presented in Table 3.2 that does not include text input allowing only for speech input. Hence, some of the systems in Table 3.2 fail to include some input and output modalities. VoiceLog (BBN 2002, Bers et al. 1998), MUST (Almeida et al. 2002), GuideShoes (Nemirovsky & Davenport 2002), The Guide (Cohen-Rose & Christiansen 2002), and QuickSet (Oviatt et al. 2000) all fail to include animation in their output. Of these, the latter three systems also fail to use speech on output. GuideShoes is the only other mobile intelligent multimedia system that outputs non-speech audio, but this is not combined with other output modalities, so it could be considered a unimodal communication.

With TeleMorph's ability on the client side to receive a variety of streaming media/modalities, TeleTuras will be able to present a multimodal output presentation including non-speech audio that will provide relevant background music about a certain tourist point of interest (e.g. theatre/concert venue). The focus with TeleMorph's output presentation lies in the chosen modalities and the rules and constraints that determine these choices. TeleMorph will implement a comprehensive set of input and output modalities. On input TeleMorph will handle text, speech and Haptic modalities, whilst output will consist of text, Text-To-Speech (TTS), non-speech audio, static graphics and animation. This will provide output similar to that produced in most current intelligent multimedia systems which mix text, static graphics (including map, charts and figures) and speech (some with additional non-speech audio) modalities. Besides the systems listed in the table, many other practical applications of intelligent multimedia interfaces have been developed in domains such as intelligent tutoring, retrieving information from a large database, car-driver interfaces, real estate presentation and car exhibition.

3.7 Project schedule and current status

The work proposed here requires several steps to be carried out in order to achieve the desired objectives of TeleMorph and TeleTuras. Table B.1 in Appendix B outlines the main tasks and milestones of this project. The literature review has been completed; it included research into the following relevant areas: mobile intelligent multimedia systems, network-adaptive multimedia models, semantic representation, fusion and coordination of modalities, intelligent multimedia presentation systems, intelligent multimedia interfaces, intelligent multimedia agents, Cognitive Load Theory and Causal Probabilistic Networks. The majority of software analysis has also been completed with only a section (AI techniques) remaining to be reviewed. System design has been partially completed (sections 3.2-3.4). The software tools that have been reviewed in respect of TeleMorph are discussed in the next chapter.

Categories	Systems	NLP Component		Multimodal interaction								
		Natural language generation	Natural language understanding	Input Media				Output Media				
				Text	Pointing (haptic deixis)	Speech	Vision	Text	Audio		Visual	
									Text to speech	Non-speech audio	Graphics (static)	Animation
Intelligent Multimedia Presentation systems	WIP		●	●				●			●	
	COMET		●	●				●	●		●	
	TEXTPLAN			●	●			●	●			●
	Cicero	●	●	●	●	●		●			●	
	IMPROVISE			●							●	
Intelligent Multimedia Interfaces	AIMI			●	●	●		●	●	●	●	
	AlFresco	●	●		●	●		●		●	●	
	XTRA		●	●	●			●		●	●	
	CUBRICON	●	●	●	●	●			●	●	●	
Mobile Intelligent Multimedia systems	SmartKom	●	●	●	●	●	●	●	●		●	●
	DEEP MAP		●	●	●	●		●	●		●	●
	CRUMPET	●			●	●		●	●		●	●
	VoiceLog		●		●	●		●	●		●	
	MUST	●	●	●	●	●		●	●		●	
	GuideShoes	●	●	●	●	●				●	●	
	The Guide	●	●	●	●	●		●			●	
	QuickSet	●		●	●	●		●			●	
Intelligent Multimodal Agents	Cassell's SAM & Rea (BEAT)	●	●	●		●	●		●			●
	Gandalf		●			●	●		●			●
This Project	TeleMorph &TeleTuras	●	●	●	●	●		●	●	●	●	●

Table 3.2 Comparison of Intelligent Multimedia Systems

4. Software analysis

Rather than trying to build TeleMorph from scratch, existing software tools will be made use of for speech recognition, Text-To-Speech (TTS), autonomous agent output, playing media presentations, attaining client device information, networking, bandwidth monitoring, Causal Probabilistic Networks (CPNs)/ Bayesian Belief Networks (BBNs) and streaming media. An analysis of reusable development tools for TeleMorph began at an early stage of the project and most have been decided upon. Several potential tools have been identified and analysed. Software tools for developing the TeleMorph client output module are described initially, then those to be used in the development of the TeleMorph client input are described, followed by the software tools to be reused in the implementation of the TeleMorph server.

4.1 Client output

Output on thin client devices connected to TeleMorph will primarily utilise a SMIL media player which will present video, graphics, text and speech to the end user of the system. The J2ME Text-To-Speech (TTS) engine will also be used to process speech output to the user. An autonomous agent will also be integrated into the TeleMorph client for output as they serve as an invaluable interface agent to the user as they incorporate modalities that are the natural modalities of face-to-face communication among humans.

4.1.1 SMIL media players

As portrayed in Figure 3.1, a *Media Player* will be used on the client side of TeleMorph to display the multimodal (animations, graphics, text, audio) presentation being received from the *Streaming Server*. It was decided to reuse a SMIL media player, as the development of a media player specific to TeleMorph would be outside the remit of this research and too time consuming. Some SMIL 2.0 specification based players that are currently available include:

- AMBULANT Open Source SMIL Player by CWI.
- RealOne Platform by RealNetworks, which has full support for the SMIL 2.0 Language profile.
- GRiNS for SMIL-2.0 by Oratrix is a SMIL 2.0 player that supports SMIL 2.0 syntax and semantics.
- SMIL Player by InterObject. This player supports the SMIL 2.0 Basic Profile. The player runs on PC with Windows NT/2000/XP and handheld devices with Pocket PC, such as Compaq iPAQ.
- Internet Explorer 6.0 by Microsoft includes implementation of XHTML+SMIL Profile Working Draft
- Internet Explorer 5.5 by Microsoft supports many of the SMIL 2.0 draft modules including Timing and Synchronization, BasicAnimation, SplineAnimation, BasicMedia, MediaClipping, and BasicContentControl.
- NetFront v3.0 is a micro browser for PDA/mobile phone/information appliances. It supports HTML 4.01/XHTML 1.0/ SMIL Basic/SVG Tiny.
- Pocket SMIL developed by INRIA is a SMIL 2.0 Player for the Pocket PC written in C++.

- RubiC is developed by Roxia Co.,Ltd. It includes an authoring tool and player, and fully supports SMIL 2.0 specification. "RubiC" is also available for mobile handset for mobile internet MMS(Multimedia Messaging Service)
- TAO's announced Qi browser supports SMIL, HTML 4.01 CSS, and XML (including XML Parser, DTD and Schema validation).

Most of these SMIL players also include an edit/development tool for creating SMIL presentations. Various players have been identified and investigated in this project to determine their reusability and usefulness within the context of the TeleMorph architecture. Players will be analysed further to this to ensure compatibility during development of TeleMorph.

4.1.2 J2ME programming environment

J2ME (Java 2 Micro Edition) is an ideal programming language for developing TeleMorph & TeleTuras, as it is the target platform for the Java Speech API (JSAPI) (JCP 2002). The JSAPI allows developers to incorporate speech technology into user interfaces for their Java programming language applets and applications. This will provide the utilities for Speech recognition and TTS which TeleMorph will use in its input and output respectively. Even though J2ME has graphics APIs these will not be reused for client output, as the SMIL media player will handle graphical and text output modalities.

4.1.3 Speech output - TTS

As mentioned in section 4.1.1 above, a SMIL media player will output audio on the client device. This audio will consist of audio files that are streamed to the client when the necessary bandwidth is available. But when sufficient bandwidth is unavailable audio files will be replaced by ordinary text which will be processed by a TTS engine on the client producing synthetic speech output. The Java Speech API Markup Language (JSML & JSGF 2002) is a companion specification to the Java Speech API. JSML (currently in beta) defines a standard text format for marking up text for input to a speech synthesiser. JSAPI does not provide any speech functionality itself, but through a set of APIs and event interfaces, access to speech functionality (provided by supporting speech vendors) is accessible to the application. As it is inevitable that a majority of tourists will be foreigners it is necessary that TeleTuras can process multilingual speech synthesis. To support this an IBM implementation of JSAPI "speech for Java" will be utilised. It supports US and UK English, French, German, Italian, Spanish, and Japanese. This implementation of the JSAPI is based on ViaVoice. The relationship between the JSAPI speech synthesiser and ViaVoice is shown in Figure 4.1.

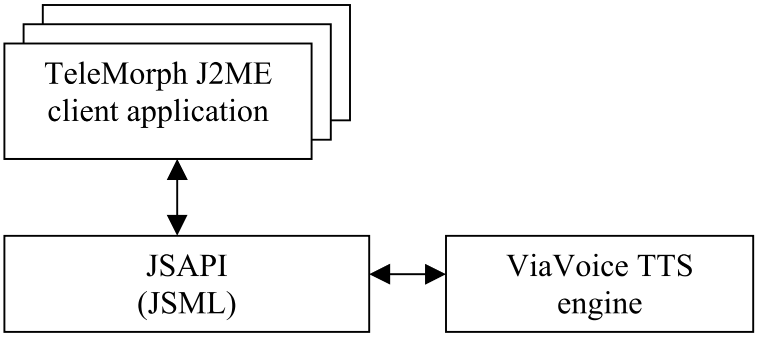


Figure 4.1 JSAPI TTS interaction with ViaVoice

4.1.4 Autonomous agents in TeleTuras

An autonomous agent will serve as an invaluable interface agent to the user as they incorporate modalities that are the natural modalities of face-to-face communication among humans. In TeleTuras it will prove very useful in communicating information on a navigation aid for tourists about sites, points of interest, and route planning. An autonomous agent in TeleMorph would be incorporated into the *Media Player* module shown in Figure 3.1.

Microsoft Agent (MS Agent 2002) provides a set of programmable software services that supports the presentation of interactive animated characters within the Microsoft Windows. It enables developers to incorporate conversational interfaces, that leverages natural aspects of human social communication. In addition to mouse and keyboard input, Microsoft Agent includes support for speech recognition so applications can respond to voice commands. Characters can respond using synthesised speech, recorded audio, or text in a cartoon word balloon. One advantage of agent characters designed by Microsoft Agent is they provide higher-levels of a character's movements often found in the performance arts, like blink, look up, look down, and walk. BEAT, another animator's tool which was incorporated in Cassell's REA, allows animators to input typed text that they wish to be spoken by an animated human figure. These tools could be used to implement actors in TeleTuras.

4.2 Client input

The TeleMorph client will allow for speech recognition, text and haptic deixis (touch screen) input. A speech recognition engine will be reused to process speech input from the user. Text and haptic input will be processed by the J2ME graphics API.

4.2.1 JSAPI J2ME speech recognition

Speech recognition in TeleMorph will be developed within *Capture Input* shown in Figure 3.1. The Java Speech API Markup Language (JSML & JSGF 2002) is a companion specification to the Java Speech API similar to the JSGF specification mentioned previously (section 4.1.3). JSML defines a standard text format for marking up text for input to a speech synthesiser. As mentioned before JSAPI does not provide any speech functionality itself, but through a set of APIs and event interfaces, access to speech functionality (provided by supporting speech vendors) is accessible to the application. For this purpose IBM's implementation of JSAPI "speech for Java" will be utilised to provide multilingual speech recognition functionality. This implementation of the JSAPI is based on ViaVoice, which will be positioned remotely in the *Interaction Manager* module on the server. The relationship between the JSAPI speech recogniser (in the *Capture Input* module in Figure 3.1) on the client and ViaVoice (in the *Interaction Manager* module in Figure 3.1) on the server is necessary as speech recognition is computationally too heavy to be processed on a thin client. After the ViaVoice speech recogniser has processed speech which is input to the client device, it will also need to be analysed by an *NLP* module to assess its semantic content. A reusable tool to do this is yet to be decided upon to complete this task. Possible solutions for this include adding an additional NLP component to ViaVoice; or perhaps reusing other natural understanding tools such as PC-PATR (McConnel 1996) which is a natural language parser based on context-free phrase structure grammar and unifications on the feature structures associated with the constituents of the phrase structure rules.

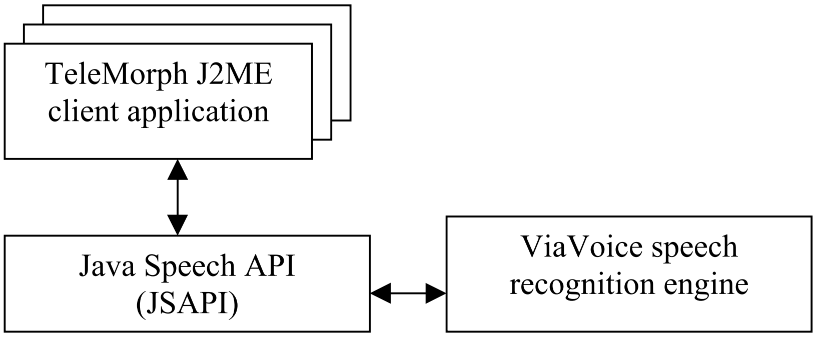

TeleMorph J2ME client application
Java Speech API (JSAPI)
ViaVoice speech recognition engine

Figure 4.2 JSAPI speech recogniser interaction with ViaVoice

4.2.2 J2ME graphics APIs

The User Interface (UI) defined in J2ME is logically composed of two sets of APIs, High-level UI API which emphasises portability across different devices and the Low-level UI API which emphasises flexibility and control. The portability in the high-level API is achieved by employing a high level of abstraction. The actual drawing and processing user interactions are performed by implementations. Applications that use the high-level API have little control over the visual appearance of components, and can only access high-level UI events. On the other hand, using the low-level API, an application has full control of appearance, and can directly access input devices and handle primitive events generated by user interaction. However the low-level API may be device-dependent, so applications developed using it will not be portable to other devices with a varying screen size. TeleMorph will use a combination of these to provide the best solution possible. Using these graphics APIs TeleMorph will implement the *Capture Input* module which will accept text from the user. Also using these APIs, haptic input will be processed by the *Capture Input* module to keep track of the user’s input via a touch screen, if one is present on the device. User preferences in relation to modalities and cost incurred will also be implemented into the TeleMorph *Capture Input* module in the form of standard check boxes and text boxes available in the J2ME high level graphics API.

4.2.3 J2ME networking

J2ME networking APIs will be implemented using sockets in the *Networking API* module as shown in Figure 3.1 to communicate data from the *Capture Input* module to the *Media Analysis* and *Constraint Information Retrieval* modules on the server. Information on client device constraints will also be received from the *Device Monitoring* module to the *Networking API* and sent to the relevant modules within the *Constraint Information Retrieval* module on the server. Networking in J2ME has to be very flexible to support a variety of wireless devices and has to be device specific at the same time. To meet this challenge, the Generic Connection Framework (GCF) is incorporated into J2ME. The idea of the GCF is to define the abstractions of the networking and file input/output as generally as possible to support a broad range of devices, and leave the actual implementations of these abstractions to the individual device manufacturers. These abstractions are defined as Java interfaces. The device manufacturers choose which one to implement based on the actual device capabilities. The *sysInfo* tool described in the next section will retrieve information on the device's networking capabilities as necessary. The GCF supports five basic forms of communications: HTTP, Sockets, Datagrams, Serial Port and Files. Wireless networking in the TeleMorph client will utilise sockets for communicating user input to the TeleMorph server.

4.3 Client device status

The SysInfo (2003) J2ME application (or MIDlet) will be used for easy retrieval of a device's capabilities in the *Device Monitoring* module as shown in Figure 3.1. It probes several aspects of the J2ME environment it is running in and lists the results. In particular, it will try to establish a networking connection to find out which protocols are available, check device memory, the Record Management System (RMS) and other device properties. The following are an explanation of the various values collected by the MIDlet.

Properties

Contains basic properties that can be queried via System.getProperty(). They reflect the configuration and the profiles implemented by the device as well as the current locale and the character encoding used. The *platform* property can be used to identify the device type, but not all vendors support it.

Memory

Displays the total heap size that is available to the Java virtual machine as well as the flash memory space available for RMS. The latter value will depend on former RMS usage of other MIDlets in most cases, so it doesn't really reflect the total RMS space until you run SysInfo on a new or "freshly formatted" MIDP device. The MIDlet also tries to detect whether the device's garbage collector is compacting, that is, whether it is able to shift around used blocks on the heap to create one large block of free space instead of a large number of smaller ones.

Screen

Shows some statistics for the device's screen, most notably the number of colors or grayscales and the resolution. The resolution belongs to the canvas that is accessible to MIDlets, not to the total screen, since the latter value can't be detected.

Protocols

Lists the protocols that are supported by the device. HTTP is mandatory according to the J2ME MIDP specification, so this one should be available on every device. The other protocols are identified by the prefix used for them in the *Connector* class:

- http - Hypertext Transfer Protocol (HTTP)
- https - Secure Hypertext Transfer Protocol (HTTPS)
- socket- Plain Transmission Control Protocol (TCP)

- ssocket - Secure Transmission Control Protocol (TCP+TLS)
- serversocket - Allows to listen in incoming connections (TCP)
- datagram - User Datagram Protocol (UDP)
- comm - Gives access to the serial port
- file - Gives access to the device's flash memory file system

Limits

Reflects some limitations that a device has. Most devices restrict the maximum length of the TextField and TextBox classes to 128 or 256 characters. Trying to pass longer contents using the setString`()` method might result in an IllegalArgumentException being thrown, so it is best to know these limitations in advance and work around them. Also, several devices limit the total number of record stores, the number of record stores that can be open at the same time, and the number of concurrently open connections. For all items, "none" means that no limit could be detected.

Speed

The MIDlet also does some benchmarking for RMS access and overall device speed. This last section holds values gained during these benchmarks. The first four items show the average time taken for accessing an RMS record of 128 bytes using the given method. The last item shows the time it took the device to calculate the first 1000 prime numbers using a straightforward implementation of Eratosthenes' prime sieve algorithm. While this is not meant to be an accurate benchmark of the device's processor, it can give an impression of the general execution speed (or slowness) of a device and might be a good hint when to include a "Please wait" dialog.

4.4 TeleMorph server tools

The software tools to be used in the implementation of the TeleMorph server application are detailed below. SMIL will be utilised to form the semantic representation language in TeleMorph and will be processed by the *Presentation Design* module in Figure 3.1. MPEG-7 may also be integrated with SMIL but this is undecided yet. The HUGIN development environment is also described below, this environment allows TeleMorph to develop its decision making process using Causal Probabilistic Networks which will form the *Constraint Processor* module as portrayed in Figure 3.1. The ViaVoice speech recognition software is one other software tool that will be used in the server side implementation of TeleMorph within the *Interaction Manager* module. On the server end of the system a *Streaming Server* will be set up to transmit the output presentation from the TeleMorph server application to the client device's *Media Player*. The Darwin streaming server (Darwin 2003) may be used for this purpose. Also in this section a brief description is given of the JATLite middleware which will be used in the development of TeleMorph if it is considered necessary at time of implementation.

4.4.1 SMIL semantic representation

As discussed previously the XML based SMIL language will form the semantic representation language of TeleMorph which will be used in the *Presentation Design* module as shown in Figure 3.1. TeleMorph will design SMIL content that comprises multiple modalities that exploit currently available resources fully, whilst considering various constraints that affect the presentation, but in particular, bandwidth. This output presentation will then be streamed to the *Media Player* module on the mobile client for displaying to the end user. TeleMorph will constantly recycle the presentation SMIL code to adopt to continuous and unpredictable variations of physical system constraints (e.g. fluctuating bandwidth, device memory), user constraints (e.g. environment) and user choices (e.g. streaming text instead of synthesised speech). In order to present the content to the end user there will be a SMIL media player available on the client device. Section 4.1.1 discussed currently available SMIL 2.0 players. As MPEG-7, discussed in section 2.4.4, describes multimedia content using XML this may be incorporated with the SMIL language to assist with the semantic representation in TeleMorph.

4.4.2 TeleMorph reasoning - CPNs/BBNs

Causal Probabilistic Networks will be used in TeleMorph to conduct reasoning and decision making within the *Constraints Processor* module as shown in Figure 3.1. In order to implement Bayesian Networks in TeleMorph the HUGIN (HUGIN 2003, Jensen & Jianming 1995) development environment will be used. HUGIN provides the necessary tools to construct Bayesian Networks. When a network has been constructed, one can use it for entering evidence in some of the nodes where the state is known and then retrieve the new probabilities calculated in other nodes corresponding to this evidence. A Causal Probabilistic Network (CPN)/Bayesian Belief network (BBN) is used to model a domain containing uncertainty in some manner. It consists of a set of nodes and a set of directed edges between these nodes. A Belief Network is a Directed Acyclic Graph (DAG) where each node represents a random variable. Each node contains the states of the random variable it represents and a conditional probability table (CPT) or, in more general terms, a conditional probability function (CPF). The CPT of a node contains probabilities of the node being in a specific state given the states of its parents. Edges reflect cause-effect relations within the domain. These effects are normally not completely deterministic (e.g. disease -> symptom). The strength of an effect is modelled as a probability.

4.4.3 JATLite middleware

As TeleMorph is composed of several modules with different tasks to accomplish, the integration of the selected tools to complete each task is important. To allow for this a middleware may be required within the *TeleMorph Server* as portrayed in Figure 3.1.

One such middleware is JATLite (JATLite 2003, Jeon et al. 2000) which was developed by the Stanford university. JATLite provides a set of Java packages which makes it easy to build multi-agent systems using Java. JATLite features modular construction consisting of increasingly specialised layers which allows for flexibility in the infrastructure. Four different layers are incorporated to achieve this, including:

- Abstract layer- provides a collection of abstract classes necessary for JATLite implementation. Although JATLite assumes all connections to be made with TCP/IP, the abstract layer can be extended to implement different protocols such as UDP.
- Base layer- provides communication based on TCP/IP and the abstract layer. There is no restriction on the message language or protocol. The base layer can be extended, for example, to allow inputs from sockets and output to files. It can also be extended to give agents multiple message ports.
- KQML (Knowledge Query and Manipulation Language) layer- provides for storage and parsing of KQML messages.
- Router layer- provides name registration and message routing and queuing for agents via the AMR.

If it is determined during the implementation of TeleMorph that middleware is necessary then JATLite will be utilised. As an alternative to the JATLite middleware The Open Agent Arhcitecture (OAA) (Cheyer & Martin 2001) could be used. OAA is a framework for integrating a community of heterogeneous software agents in a distributed environment. Psyclone (2003) is a flexible middleware that can be used as a blackboard server for distributed, multi-module and multi-agent systems which could also be utilised.

In this section we have described various software tools that will be used in the implementation of TeleMorph for speech recognition, Text-To-Speech (TTS), autonomous agent output, playing media presentations, attaining client device information, networking, bandwidth monitoring, Causal Probabilistic Networks/Bayesian Belief Networks and streaming media. An analysis of these reusable development tools has been given in the context of the proposed design of TeleMorph (shown in Figure 3.1).

5. Conclusion and future work

In this chapter we conclude by first summarising the research proposal, the literature review and the main research areas and contribution of this project. We then discuss the software analysis focusing on tools and programming languages to be used, followed by potential applications of the research, in particular TeleTuras. Finally we give some thoughts on promising future directions.

This objective of the work described in this research plan is the development of a mobile intelligent multimedia presentation architecture called TeleMorph. TeleMorph will be able to dynamically generate a multimedia presentation from semantic representations using output modalities that are determined by constraints that exist on a mobile device's wireless connection, the mobile device itself and also those limitations experienced by the end user of the device. The output presentation will include Language and Vision modalities consisting of video, speech, non-speech audio and text. Input to the system will be in the form of speech, text and haptic deixis. The objectives of TeleMorph are: (1) receive and interpret questions from the user, (2) map questions to multimodal semantic representation, (3) match multimodal representation to knowledge base to retrieve answer, (4) map answers to multimodal semantic representation, (5) monitor user preference or client side choice variations, (6) query bandwidth status, (7) detect client device constraints and limitations, (8) generate multimodal presentation based on constraint data. The architecture, data flow, and issues in the core modules of TeleMorph such as constraint determination and automatic modality selection are also introduced in this report. A tourist information aid for the city of Derry, TeleTuras, will be developed as a testbed for TeleMorph. It will communicate TeleMorph-adapted presentations to tourists consisting of: route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. The main focus in TeleTuras will be on the output modalities used to communicate this information and also the effectiveness of this communication. Applying a collection of common questions will test TeleTuras. These questions will be accumulated by asking prospective users/tourists what they would require from a tourist information aid such as TeleTuras.

Previous research in the related areas of wireless telecommunications, mobile intelligent multimedia systems, network-adaptive multimedia models, semantic representation, fusion and coordination of modalities, intelligent multimedia presentation systems, intelligent multimedia interfaces, intelligent multimedia agents, Cognitive Load Theory (CLT) and Causal Probabilistic Networks (CPNs) have been reviewed. Most of the current work on mobile intelligent multimedia presentation systems determine the multimodal output presentation by considering constraints imposed by the client device or the end user of the system. As mentioned earlier (in section 3.6.1) the issue of device diversity is considered by a number of current systems. Some of these systems are simply aware of the type of output device (e.g. PDA, desktop, laptop, TV) (e.g. EMBASSI) and others are concerned with the core resources that are available on the client device (e.g. memory, CPU, output capabilities) (e.g. SmartKom-mobile). Some mobile intelligent multimedia systems also allow for methods of user choice/preference when choosing output modalities in order to present a more acceptable output presentation to the end user. Some other mobile intelligent multimedia systems support awareness of the user's domain, environment and preferences. One other factor that is relevant to mobile intelligent multimedia systems is Cognitive Load Theory (CLT). This theory states the most efficient (judged by user retention) and the most appealing (user-acceptable) modalities for portraying information on various types of device (PDA, TV, Desktop). One system that takes this theory into account is EMBASSI (Hildebrand 2000). The RAJA framework which is aimed at the development of resource-adaptive multi-agent systems was intended to provide resource-aware modules for DEEP MAP but this has not been integrated yet. Hence, these systems fail to take the bandwidth of the mobile network into account when choosing output presentation modalities. One other main issue which the systems reviewed fail to consider is the resultant effect of the union of all the relevant constraints (i.e. network bandwidth, mobile device and end user constraints). TeleMorph will improve upon existing systems by dynamically morphing the output presentation modalities depending on: (1) available network bandwidth (also network latency, packet loss and error rate); (2) available client device resources (mobile device display resolution, available memory and processing power, available output abilities); (3) end user modality preferences and associated costs incurred in downloading a presentation; (4) the cognitive load of the end user (determined by Cognitive Load Theory) and whether the function of the presentation is directed towards end-user retention (e.g. a city tour) or is intended solely for informative purposes (e.g. information about an interesting sight nearby). These constraints contribute to the automatic adaptation features of TeleMorph, which will consider a union of their effects using Causal Probabilistic Networks. Hence the main area of contribution in TeleMorph is its awareness of available bandwidth and the union of this with other relevant constraints.

Accordingly, TeleMorph's unique contribution has been identified – *Bandwidth determined Mobile Multimodal Presentation*. TeleMorph will dynamically morph the output presentation between output modalities depending on available network bandwidth. This system will be an improvement on previous systems in the following ways:

- The TeleMorph server application will be aware of important network issues with primary focus on network Bandwidth. Other secondary issues that will be considered in TeleMorph include network latency and bit error rate, mobile device constraints (display, available output modalities, memory, processing power) and user constraints (modality preferences, cost incurred, cognitive load) which will be combined and utilised to determine morphing behaviour between output modalities during presentation design.
- TeleMorph's presentation design will use the Causal Probabilistic Network (CPN) approach to analyse a union of all relevant constraints imposed on TeleMorph and decide on the optimal multimodal output presentation.
- TeleMorph will provide output that adheres to good usability practice, resulting in suitable throughput of information and context sensitive modality combinations in keeping with the Cognitive Load Theory.

TeleMorph will be developed as a client-server architecture using existing software tools such as Java 2 Micro Edition (J2ME), Synchronised Multimedia Integration Language (SMIL) editors/players, HUGIN and JATLite. The J2ME programming environment provides a set of tools and Application Programming Interfaces (APIs) to develop speech (using the Java Speech API) graphics (using graphics APIs) and text (using high level graphics API) input/output modules for the client application. The graphics API in J2ME will also allow for haptic input by the user. Other J2ME tools that will be reused in the implementation of the client include the sysInfo application (for detecting device status) and wireless networking APIs (for communicating input to the server). On the client side of TeleMorph an appropriate intelligent multimedia agent (e.g. MS Agent) will be used to implement an actor for more effective communication. The SMIL language (XML based) will be used as the semantic representation language in TeleMorph whilst possibly reusing elements of the MPEG-7 standard (also XML based). To present SMIL based multimedia presentations on the mobile client device (e.g. Compaq iPaq) a SMIL compatible media player will be used. TeleMorph's server-side presentation design module will use the HUGIN development environment which is based on the Causal Probabilistic Network (CPN) approach to reasoning and decision making. HUGIN will analyse a union of all relevant constraints imposed on TeleMorph and decide on the optimal multimodal output presentation. Middleware such as JATLite or Open Agent Architecture will provide a solution for integration and interoperation between TeleMorph's various modules.

There are a number of problems that have to be solved before TeleMorph can be widely applied. To accomplish intelligent presentation design whilst considering various relevant constraints the reasoning and decision making technique, Causal Probabilistic Networks (CPNs), employed by TeleMorph will be investigated further. This will consist of modeling TeleMorph's domain variables (constraint information) and probabilistic dependencies between these variables (i.e. cause-effect relations within the domain). This element of TeleMorph will be combined with work on constraint retrieval techniques, including bandwidth data, client device status and user preference data. An apt mobile client device emulator will be chosen to use for design, implementation and testing of the TeleMorph architecture. The module related to input capture on the client will be designed enabling various input modalities. This module will be integrated with a SMIL media player which will deal with multimodal output. A large section of future work related to the design of presentations will be in mapping input queries to a semantic representation and designing a constraint-aware multimodal output presentation in response, whilst also ensuring coordination of modalities. The integration of TeleMorph's various modules will be achieved using middleware, which will be examined further throughout the design and implementation stages. TeleTuras will incorporate the solution provided by TeleMorph and will communicate TeleMorph-adapted presentations designed for the tourism domain. Following the development of TeleMorph, the implementation of TeleTuras will ensue, which will demonstrate TeleMorph's effectiveness.

References

Almeida, L., I. Amdal, N. Beires, M. Boualem, L. Boves, E. den Os, P. Filoche, R. Gomes, J.E. Knudsen, K. Kvale, J. Rugelbak, C. Tallec & N. Warakagoda (2002) The MUST guide to Paris - Implementation and expert evaluation of a multimodal tourist guide to Paris. In Proc. ISCA Tutorial and Research Workshop (ITRW) on Multi-Modal Dialogue in Mobile Environments (IDS 2002), 49-51, Kloster Irsee, Germany, June 17-19.

Almeida, L., I. Amdal, N. Beires, M. Boualem, L. Boves, E. den Os, P. Filoche, R. Gomes, J.E. Knudsen, K. Kvale, J. Rugelbak, C. Tallec & N. Warakagoda (2002) Implementing and evaluating a multimodal and multilingual tourist guide. International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems Copenhagen, Denmark, 28-29 June.

Amir, E., S. McCanne, & H. Zhang (1995) An Application Level Video Gateway. In ACM Multimedia '95. San Francisco, CA, Nov. 9-13, 255-265.

André, E., J. Müller and T. Rist (1996) The PPP Persona: A Multipurpose Animated Presentation Agent. In Advanced Visual Interfaces, T. Catarci, M.F. Costabile, S. Levialdi and G. Santucci (Eds.), 245-247, New York, USA: ACM Press.

André, E. and T. Rist (2000) 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), Los Angeles, 1-8.

André, E., T. Rist, S. van Mulken, M. Klesen, & S. Baldes (2000) The Automated Design of Believable Dialogues for Animated Presentation Teams, In J. Cassell, S. Prevost, J. Sullivan, and E. Churchill, Eds., 220-255, Embodied Conversational Agents, The MIT Press.

André, E. & T. Rist (2001) Controlling the Behavior of Animated Presentation Agents in the Interface: Scripting vs. Instructing. In: *AI Magazin*e, Vol. 22, No. 4, 53-66.

Apple (2003)
Quicktime online http://www.apple.com.quicktime/ Site visited 30/09/2003

Arens, Y., E.H. Hovy, & S. Van Mulken (1993) Structure and Rules in Automated Multimedia Presentation Planning. In Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93), Vol. 2, 1253--1259, Chambery, France, September.

Arens, Y. & E.Y. Hovy (1995) The Design of a Model-based Multimedia Interaction Manager. In Integration of Natural Language and Vision Processing, Vol II, Intelligent Multimedia, P. Mc Kevitt (Ed.), 95-115. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Baddeley, A. D. & R.H. Logie (1999) Working Memory: The Multiple-Component Model. In Miyake, A. and Shah, P. (Eds.), 28-61, Models of working memory: Mechanisms of active maintenance and executive control, Cambridge University Press.

Barrett, R., Maglio P.P., & Kellem D.C. (1997) WBI - a confederation of agents that personalise the Web. In Proceedings of the First International Conference on Autonomous Agents, New York, NY: ACM Press, 496-499.

BBN (2002)
http://www.bbn.com/ Site visited 30/09/2003

Bellifemine, F., A. Poggi, G. Rimassa (1999) JADE - A FIPA-compliant agent framework. CSELT internal technical report. Part of this report has been also published in Proceedings of PAAM'99, The Commonwealth Conference and Events Centre, Kensington, London, April, 97-108.

Berners-Lee, T., J. Hendler, & O. Lassila (2001) The Semantic Web, Scientific American, May 17. (URLhttp://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2).

Bers, J., S. Miller & J. Makhoul (1998) Designing Conversational Interfaces with Multimodal Interaction. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne Conference Resort, Lansdowne, Virginia, 88-92.

Bolt, R.A. (1987) Conversing with Computers. In Readings in Human-Computer Interaction: A Multidisciplinary Approach, R. Baecker and W. Buxton (Eds.), California: Morgan-Kaufmann.

Brewer, E.A., R.H. Katz, Y. Henderson, E. Amir, H. Balakrishnan, A. Fox, V. Padmanabhan, & S. Seshan (1998) A network Architecture for Heterogeneous Mobile Computing. IEEE Personal Communications Magazine, Vol. 5 (No. 5), 8-24.

Brøndsted, T., P. Dalsgaard, L.B. Larsen, M. Manthey, P. Mc Kevitt, T.B. Moeslund & K.G. Olesen (1998) A platform for developing Intelligent MultiMedia applications. Technical Report R-98-1004, Center for PersonKommunikation (CPK), Institute for Electronic Systems (IES), Aalborg University, Denmark, May.

Brøndsted, T., P. Dalsgaard, L.B. Larsen, M. Manthey, P. Mc Kevitt, T.B. Moeslund and K.G. Olesen (2001) The IntelliMedia WorkBench - An Environment for Building Multimodal Systems. In Advances in Cooperative Multimodal Communication: Second International Conference, CMC'98, Tilburg, The Netherlands, January 1998, Selected Papers, Harry Bunt and Robbert-Jan Beun (Eds.), 217-233. Lecture Notes in Artificial Intelligence (LNAI) series, LNAI 2155, Berlin, Germany: Springer Verlag.

Bunt, H. & L. Romary (2002) Towards multimodal content representation. In Proceedings of International Standards of Terminology and Language Resources Management, LREC 2002, Las Palmas, Spain, 54-60.

Burger, J. & R. Marshall (1993) The Application of Natural Language Models to Intelligent Multimedia. In Intelligent Multimedia Interfaces, M. Maybury (Ed.), 167-187, Menlo Park: AAAI/MIT Press.

Cassell, J., J. Sullivan, S. Prevost & E. Churchill, (Eds.) (2000) Embodied Conversational Agents. Cambridge, MA: MIT Press.

Cassell, J., H. Vilhjalmsson & T. Bickmore (2001) BEAT: the Behavior Expression Animation Toolkit, Computer Graphics Annual Conference, SIGGRAPH 2001 Conference Proceedings, Los Angeles, Aug 12-17, 477-486.

CCPP (2003)
http://www.w3.org/TR/NOTE-CCPPexchange.html Site visited 30/09/03

Cheyer, A. & Martin, D. (2001) The Open Agent Architecture. Journal of Autonomous Agents and Multi-Agent Systems, Vol. 4, No. 1, March, 143-148.

Cohen-Rose, A.L. & S.B. Christiansen (2002) The Hitchhiker's Guide to the Galaxy. In Language Vision and Music, P. Mc Kevitt, S. Ó Nualláin and C. Mulvihill (Eds.), 55-66, Amsterdam: John Benjamins.

Constantiou, I.D., N. Mylonopoulos & G. Polyzos (2001) Forth Generation Networks and Interconnection Issues. Proceedings of Wireless World Research Forum (WWRF) Kick off Meeting, Munich, Germany.

Coyne, B. & R. Sproat (2001) WordsEye: An Automatic Text-to-Scene Conversion System. Computer Graphics Annual Conference, SIGGRAPH 2001 Conference Proceedings, Los Angeles, Aug 12-17, 487- 496.

Crumpet (2002)
http://www.ist-crumpet.org Site visited 30/09/2003

DAML (2003)
DAML homepage http://www.daml.org/ Site last visited 30/09/2003

Darwin (2003)
http://developer.apple.com/darwin/projects/darwin/ Site visited 30/09/2003

Ding, Y., C. Kray, R. Malaka, & M. Schillo (2001) RAJA - A Resource-adaptive Java Agent Infrastructure. Proceedings of the Fifth International Conference on Autonomous Agents (AA'01), Montreal, Canada, May 28 - June 1, 332-340.

Elsen, I., F. Hartung, U. Horn, M. Kampmann, & L. Peters (2001) Streaming Technology in 3G Mobile Communication Systems. In IEEE Computer, 34(9), September, 46-52.

Elting, C., J. Zwickel & R. Malaka (2001) Device-Dependant Modality Selection for User-Interfaces - An Empirical Study. International Conference on Intelligent User Interfaces. IUI 2002, San Francisco, CA.

EML (2002)
http://www.eml.org/english/Research/Memory Site visited 30/09/2003.

Feiner, S.K. & K.R. McKeown (1991a) COMET: Generating coordinated multimedia explanations. In S.P. Robertson, G. M. Olson, and J. S. Olson (Eds.), 449-450, CHI'91 Conference Proceedings, Reading, MA:Addison-Wesley.

Feiner, S.K. & K.R. McKeown (1991b) Automating the Generation of Coordinated Multimedia Explanations. IEEE Computer, 24(10): 33-41.

Fell, H., H. Delta, R. Peterson, L. Ferrier, Z. Mooraj, M. Valleau (1994) Using the baby-babble-blanket for infants with motor problems. Proceedings of the Conference on Assistive Technologies (ASSETS'94), 77-84. Marina del Rey, CA.
(URL http://www.acm.org/sigcaph/assets/assets98/assets98index.html)

Fink, J. & A. Kobsa (2002) User modeling for personalised city tours. Artificial Intelligence Review, 18(1) 33–74.

FIPA-OS (2003)
http://www.nortelnetworks.com/fipa-os Site visited 30/09/03

FIPA (2003)
Foundation for Intelligent Physical Agents Specifications http://www.fipa.org Site visited 30/09/03

Fox, A., & E.A. Brewer (1996) Reducing WWW Latency and Bandwidth Requirements by Real-Time Distillation. In Proceedings of Fifth International World Wide Web Conference (WWW-5), Paris, France, May, 1444-1456.

Fox, A., S.D. Gribble, E.A. Brewer, & E. Amir. (1996) Adapting to Network and Client Variability via On-Demand Dynamic Distillation. In Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOSVII), Cambridge, MA, October, 160-170.

Hildebrand, A. (2000) EMBASSI: Electronic Multimedia and Service Assistance. In Proceedings IMC'2000, Rostock-Warnem¨unde, Germany, November, 50-59.

Hokimoto, A. & T. Nakajima (1997) Robust host mobility supports for adaptive mobile applications. In proceedings of the international conference on world wide computing and its applications, Masuda, T., Y. Masunaga, & M. Tsukamoto (Eds.), 106-121, Springer, Berlin.

Holzman, T.G. (1999) Computer-human interface solutions for emergency medical care. Interactions, *6(3),* 13-24.

Horrocks, I. (2002) DAML+OIL: a description logic for the semantic web. Bull. of the IEEE Computer Society Technical Committee on Data Engineering, 25(1):4-9, March.

Horrocks, I., D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, & E. Motta (2000) OIL: The Ontology Inference Layer. Technical Report IR-479, Vrije Universiteit Amsterdam, Faculty of Sciences, September.
(http://www.ontoknowledge.org/oil/)

Horrocks, I., P.F. Patel-Schneider & F.V. Harmelen (2002) Reviewing the design of DAML+OIL: An ontology language for the semantic web. In Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002), Edmonton, Canada, 792-797, Menlo Park, CA, USA: AAAI Press.

Hovy, E. & Y. Arens (1993) The Planning Paradigm Required for Automated Multimedia Presentation Planning. In Human-Computer Collaboration: Reconciling Theory, Synthesising Practice. Papers from the 1993 Fall Symposium Series, Raleigh, NC, AAAI Technical Report FS-93-05.

HUGIN (2003)
http://www.hugin.com/ Site visited 30/09/2003

JATLite (2003)
http://java.stanford.edu/ Site visited 30/09/2003

Jensen, F.V. & Jianming, L. (1995) Hugin: a system for hypothesis driven data request. In Probabilistic Reasoning and Bayesian Belief Networks, Gammerman (ed.), 109-124, London, England: Alfred Waller ltd.

Jeon, H., C. Petrie & M.R. Cutkosky (2000) JATLite: A Java Agent Infrastructure with Message Routing. IEEE Internet Computing Vol. 4, No. 2, Mar/Apr, 87-96.

JCP (2002)
Java Community Process. http://www.jcp.org/en/home/index Site visited 30/09/2003

Joshi, A. (2000) On Proxy Agents, Mobility, and Web Access. ACM/Baltzer Mobile Networks and Applications Journal, 5:4, 233-241.

JSML & JSGF (2002)
Java Community Process. http://www.jcp.org/en/home/index Site visited 30/09/2003

Karshmer, A.I. & M. Blattner (Eds) (1998) Proceedings of the ACM Conference on Assistive Technologies (ASSETS'98), Marina del Rey, CA.
(URL http://www.acm.org/sigcaph/assets/assets98/assets98index.html).

Katz, R.H., E.A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. Nguyen, V. Padmanabhan & M. Stemm (1996) The Bay Area Research Wireless Access Network. Proceedings of the forty-first IEEE Spring Computer Conference (COMPCON), Santa Clara, CA, February, 15-20.

Kleinrock, L. (1995) Nomadic Computing. Select Proceedings of Third INFORMS Telecommunications Conference, Boca Raton, Florida, Volume 7, Nos. 1-3, 5-15, June.

Koch, U.O. (2000) Position-aware Speech-enabled Hand Held Tourist Information System. Semester 9 Project Report, Institute of Electronic Systems, Aalborg University, Denmark.

Liljeberg, M., H. Helin, M. Kojo, & K. Raatikainen (1996a) Enhanced Services for World Wide Web in Mobile WAN Environment. Department of Computer Science, University of Helsinki, Series of Publications C, No. C-1996-28. April 1996.

Liljeberg, M., H. Helin, M. Kojo, & K. Raatikainen (1996b) Enhanced Services for World Wide Web in Mobile WAN Environment. Univ. of Helsinki CS Tech Report C-1996-28; in Proc. IMAGE'COM 96, May 20-24, Bordeaux, France, 119-124.

Malaka, R. (2000) Artificial Intelligence Goes Mobile - Artificial Intelligence in Mobile Systems 2000, Workshop in conjunction with ECAI 2000, Berlin, Germany, August 22, Workshop Notes, 5-6.

Malaka, R. (2001) Multi-modal Interaction in Private Environments. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany, 29 October - 2 November.
(http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/Dagstuhl-2001.pdf Site visited 11/12/02)

Malaka, R. & A. Zipf (2000) DEEP MAP - Challenging IT Research in the Framework of a Tourist Information System. In: D. Fesenmaier, S. Klein, and D. Buhalis, (eds.), 15-27, Information and Communication Technologies in Tourism 2000, Proceedings of ENTER 2000, 7th International Congress on Tourism and Communications Technologies in Tourism, Barcelona (Spain), Springer Computer Science, Wien, New York.

Malaka, R., R. Porzel & A. Zipf, (2000) Integration of Smart Components for Building Your Personal Mobile Guide In: Artificial Intelligence in Mobile Systems - AIMS2000, R. Malaka, (ed.), 22-26, Workshop in conjunction with ECAI 2000, Berlin, Germany, August 22, Workshop Notes.

Mann, W.C., C.M. Matthiessen and S.A. Thompson (1992) Rhetorical Structure Theory and Text Analysis. In Discourse Description: Diverse linguistic analyses of a fund-raising text, W.C. Mann and S.A. Thompson (Eds.), 39-78, Amsterdam: John Benjamins.

Maybury, M.T. (Ed.) (1993) Intelligent Multimedia Interfaces. Menlo Park: AAAI/MIT Press.

Maybury, M.T. (Ed.) (1993) Planning Multimedia Explanations Using Communicative Acts. In Intelligent Multimedia Interfaces. Menlo Park: AAAI/MIT Press, 59-74.

Maybury, M.T. (1994) Research in Multimedia Parsing and Generation. In Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia, P. Mc Kevitt (Ed.), 31-55, London, U.K.: Kluwer Academic Publishers.

Maybury, M.T. (1995) Research in Multimedia Parsing and Generation. In Integration of Natural Languageand Vision Processing (Volume II): Intelligent Multimedia, P. Mc Kevitt (Ed.), 31-55, London, U.K.: Kluwer Academic Publishers.

Maybury, M.T. and W. Wahlster (1998) Planning Multimedia Explanations Using Communicative Acts. In Readings in Intelligent User Interfaces, M.T. Maybury & W. Wahlster (Eds.), 99-106, San Francisco, CA.: Morgan Kaufmann Press.

Maybury, M.T. (1999) Intelligent User Interfaces: An Introduction. Intelligent User Interfaces, 3-4 January 5-8, Los Angeles, California, USA.

McConnel, S. (1996) KTEXT and PC-PATR: Unification based tools for computer aided adaptation. In H. A. Black, A. Buseman, D. Payne and G. F. Simons (Eds.), Proceedings of the 1996 general CARLA conference, November 14-15, 39-95. Waxhaw, NC/Dallas: JAARS and Summer Institute of Linguistics.

Mc Kevitt, P. (Ed.) (1995a) Integration of Natural Language and Vision Processing (Volume I): Computational Models and Systems. London, U.K.: Kluwer Academic Publishers.

Mc Kevitt, P. (Ed.) (1995b) Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia. London, U.K.: Kluwer Academic Publishers.

Mc Kevitt, P. (Ed.) (1996a) Integration of Natural Language and Vision Processing (Volume III): Theory and grounding representations. London, U.K.: Kluwer Academic Publishers.

Mc Kevitt, P. (Ed.) (1996b) Integration of Natural Language and Vision Processing (Volume IV): Recent Advances. London, U.K.: Kluwer Academic Publishers.

Mc Kevitt, P. (2003) “MultiModal semantic representation”, In Proceedings of the SIGSEM Working Group on the Representation of MultiModal Semantic Information, First Working Meeting, Fifth International Workshop on Computational Semantics (IWCS-5), Harry Bunt, Kiyong Lee, Laurent Romary, and Emiel Krahmer (Eds.), Tilburg University, Tilburg, The Netherlands, January.

Mc Kevitt, P., S. Ó Nualláin and C. Mulvihill (Eds.) (2002) Language, vision and music, Readings in Cognitive Science and Consciousness. Advances in Consciousness Research, AiCR, Vol. 35. Amsterdam, Netherlands: John Benjamins Publishing.

Mc Tear, M.F. (2002) Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys, Vol. 34(1), 90-169.

Microsoft (2003) Internet Explorer 6
http://www.microsoft.com/windows/ie/default.asp Site visited 30/09/03

Minsky, M. (1975) A Framework for representing knowledge. In Readings in Knowledge Representation, R. Brachman and H. Levesque (Eds.), Los Altos, CA: Morgan Kaufmann. 245-262.

MPEG7 (2003) MPEG-7 Overview v.9, ISO/IEC JTC1/SC29/WG11, March 2003.
(URL http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm)

MS Agent (2002)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msagent/agentstartpage_7gdh.asp
Site visited 30/09/2003

Neal, J. and S. Shapiro (1991) Intelligent Multi-Media Interface Technology. In Intelligent User Interfaces, J. Sullivan and S. Tyler (Eds.), 11-43, Reading, MA: Addison-Wesley.

Nemirovsky, P. & G. Davenport (2002) Aesthetic forms of expression as information delivery units. In Language Vision and Music, P. Mc Kevitt, S. Ó Nualláin and C. Mulvihill (Eds.), 255-270, Amsterdam: John Benjamins.

Noble, B.D., M. Satyanarayanan, D. Narayanan, J.E. Tilton, J. Flinn, & K.R Walker (1997) Agile Application-Aware Adaptation for Mobility. In Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles, St. Malo, France, October, 276-287.

Okada, N. (1996) Integrating vision, motion, and language through mind. In Integration of Natural Language and Vision Processing (Volum IV): Recent Advances. McKevitt, P. (Ed.) 55-79. Dordrecht, The Netherlands: Kluwer-Academic Publishers.

Oratrix (2003)
Oratrix homepage http://www.oratrix.com/ Site visited 30/09/2003

Oviatt, S.L., Cohen, P.R., Wu, L., Vergo, J., Duncan, E., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson J., & Ferro, D. (2000) Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions, Human Computer Interaction, Vol. 15, 26 3-322. (to be reprinted in J. Carroll (ed.) Human-Computer Interaction in the New Millennium, Addison-Wesley Press: Boston, to appear in 2001)

OWL (2003)
W3C OWL overview http://www.w3.org/TR/owl-features/ Site visited 30/09/2003

Pedersen, J.S. & S.R. Larsen (2003) A pilot study in modality shifting due to changing network conditions. MSc Thesis, Center for PersonCommunication, Aalborg University, Denmark.

Pfisterer, D. (2001) Resource-Aware Multi-Fidelity Video Streaming. Diploma thesis, Department of Information Technology, University of Applied Sciences Mannheim, Germany.

Pieraccini, R., (2002) Wireless Multimodal – the Next Challenge for Speech Recognition. ELSNews, summer 2002, ii.2, Published by ELSNET, Utrecht, The Netherlands.

Psyclone (2003)
http://www.mindmakers.org/architectures.html Site visited 30/09/03

RealNetworks (2003)
http://www.real.com/ Site visited 30/09/03

Realplayer (2003)
http://www.real.com/player Site visited 30/09/03

RealNetworks MediaServer (2003)
http://www.realnetworks.com/products/media_delivery.html Site visited 30/09/03

Reithinger, N. (2001) Media coordination in SmartKom. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany, 29 October - 2 November. (http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/Media_Coordination_In_SmartKom/index.html Site visited 11/12/2002)

Reithinger, N., C. Lauer & L. Romary (2002) MIAMM - Multidimensional Information Access using Multiple Modalities. International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, Copenhagen, Denmark, 28-29 June.

Rickel, J., N.B. Lesh, C. Rich, C.L. Sidner & A. Gertner (2002) Collaborative Discourse Theory as a Foundation for Tutorial Dialogue. International Conference on Intelligent Tutoring Systems (ITS), Biarritz, France and San Sebastian, Spain, June, Vol. 2363, 542-551.

Rist, T. (2001) Media and Content Management in an Intelligent Driver Support System. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany, 29 October - 2 November.
(http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/rist-dagstuhl.pdf, Site visited 11/12/2002)

Romary, L. (2001) Working group on multimodal meaning representation. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany, 29 October–2 November. (http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/WG_4_Multimodal_Meaning_Representation/index.html Site visited 11/12/02)

Rutledge, L. (2001) SMIL 2.0: XML For Web Multimedia. In IEEE Internet Computing, Sept-Oct, 78-84.

Rutledge, L., & P. Schmitz (2001) Improving Media Fragment Integration in Emerging Web Formats. In Proceedings of the International Conference on Multimedia Modelling (MMM01). CWI, Amsterdam, The Netherlands, November 5-7, 147-166.

SALT (2003)
http://saltforum.org Sites visited 30/09/2003.

Seshan, S., Stemm, M., & Katz, R. (1997) Spand: Shared passive network performance discovery. In Proceedings of the 1st USENIX Symposium on Internet Technologies and Systems (USITS '97), Monterey, CA, December, 135-146.

SmartKom Consortium (2003)
SmartKom homepage http://www.smartkom.org Sites visited 30/09/2003

SMIL (2003a)
SMIL 1.0 WWW Consortium Recommendation,June 1998. http://www.w3.org/TR/REC-smil/ Site visited 30/09/2003

SMIL (2003b)
http://www.w3.org/AudioVideo/ Site visited 30/09/2003

Stock, O. and the AlFresco Project Team (1993) AlFresco: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration. In Intelligent Multimedia Interfaces, M. Maybury (Ed.), 197-224, Menlo Park: AAAI/MIT Press.

Sweller, J., J.J.G. van Merrienboer & F.G.W.C. Paas (1998) Cognitive Architecture and Instructional Design. Educational Psychology Review, 10, 251-296.

SysInfo (2003)
http://kissen.cs.uni-dortmund.de:8080/devicedb/index.html Site last visited 30/09/03

Tanaka, K. (2001) The Mobile Communications Environment in the European Union: Systems, Regulations, and Consequences in the Past, Present, and Future. In proceedings of IEEE 2001 Conference on the History of Telecommunications 25-27 July, St. John's, Newfoundland, Canada. http://www.ieee.org/organizations/history_center/cht_papers/tanaka2.pdf Site visited 11/12/02

Tennehouse, D.L. & D.J.Wetherall (1996) Towards an Active network Architecture. Computer Communication Review, Vol. 26, No. 2, April, 5-18.

Tennehouse, D.L., J.M. Smith, W.D. Sincoskie, D.J.Wetherall, & G.J. Minden (1997) A survey of active network research. IEEE Communications Magazine, Vol. 35, No. 1, January, 80-86.

Thórisson, K. (1996) Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. Ph.D. Thesis, Media Arts and Sciences, Massachusetts Institute of Technology.

VoiceXML (2003)
http://www.voicexml.org Site visited 30/09/2003

W3C (2003)
W3C homepage. http://www.w3.org Site visited 30/09/2003

W3C XML (2003)
W3C XML Schema webpage. http://www.w3.org/XML/Schema Site visited 30/09/2003

Wahlster, W. (2003): SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. Proceedings of the Human Computer Interaction Status Conference 2003, In: Krahl, R., Günther, D. (eds) , 47-62, June, Berlin: DLR.

Wahlster, W. (1998) User and discourse models for multimodal communication. In Readings in intelligent user interfaces, M. Maybury and W. Wahlster (Eds.), 359-370, San Francisco, California: Morgan Kaufmann Publishers, Inc.

Wahlster, W.N. (2001) SmartKom A Transportable and Extensible Multimodal Dialogue System. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany, 29 October - 2 November.
(http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/SmartKom_A_Transportable_and_Extensible_Multimodal_Dialogue_System/index.html, Site visited 11/12/02)

Wahlster, W., E. André, S. Bandyopadhyay, W. Graf and T. Rist (1992) WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation. In Communication from Artificial Intelligence Perspective: Theoretical and Applied Issues, J. Slack, A. Ortony and O. Stock (Eds.), 121-143, Berlin, Heidelberg: Springer Verlag.

Wahlster, W., N. Reithinger & A. Blocher (2001a) SmartKom: Multimodal communication with a life-like character. In Proceedings of Eurospeech '01, 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, Vol. 3, 1547-1550.

Wahlster, W., N. Reithinger and A. Blocher (2001b) SmartKom: Towards Multimodal Dialogues with Anthropomorphic Interface Agents. In Proceedings of International Status Conference "Human-Computer Interaction", G. Wolf and G. Klein (Eds.), 23-34, DLR, Berlin, Germany, October 2001.

Windows MediaServer (2003)
http://www.microsoft.com/windows/windowsmedia/9series/server.aspx Site last visited 30/9/03

Zaychik, V. (2001) DAML: The DARPA Agent Markup Language
http://edge.mcs.drexel.edu/GICL/talks/DAML.pdf Site last visited 30/09/2003

Zhou, M.X. & S.K. Feiner (1997) The Presentation and Usage of a Visual Lexicon for Automated Graphics Generation. In Proceedings of International Joint Conference on Artificial Intelligence, Nagoya, Japan, August, 23-29, 1056-1062.

Zhou, M.X. & S.K. Feiner (1998) IMPROVISE: automated generation of animated graphics for coordinated multimedia presentations. In Cooperative Multimodal Communication Second International Conference (CMC'98), H. Bunt and R. Beun (Eds.), 43-63, Tilburg, The Netherlands.

Zipf, A. & R. Malaka (2001) Developing Location Based Services (LBS) for tourism - The service providers view. ENTER 2001, In Proceedings of the 8th. International Congress on Tourism and Communications Technologies in Tourism, Montreal (Canada), April 24-27.

Appendix A: SMIL synchronisation modules

AccessKeyTiming	This module defines the attribute value syntax for the begin and end attributes that allow elements to begin and end based upon the user actuating a designated access key.
BasicInlineTiming	This module defines the attributes that make up basic timing support for adding timing to XML elements.
BasicTimeContainers	This module defines basic time container elements, attributes that describe an element's display behaviour within a time container, and end conditions for time containers.
EventTiming	This module defines the attribute value syntax for begin and end attributes that allow elements to begin and end in response to an event.
ExclTimeContainers	This module includes a time container that defines a mutually exclusive set of elements, and describes interrupt semantics among these elements.
FillDefault	This module defines syntax for specifying default display behaviour for elements.
MediaMarkerTiming	This module defines the attribute value syntax for the begin and end attributes that allow elements to begin and end based upon markers contained in the source content.
MinMaxTiming	This module defines the attributes that allow setting minimum and maximum bounds on element active duration.
MultiArcTiming	This module extends the attribute value syntax for the begin and end attributes to allow multiple semicolon-separated values. Any combination of the simple begin and end value types provided by the other timing modules included in the profile are allowed
RepeatTiming	This module defines the attributes that allow repeating an element for a given duration or number of iterations.
RepeatValueTiming	This module defines the attribute value syntax for begin and end attributes that allow elements to begin and end in response to repeat events with a specific iteration value.
RestartDefault	This module defines syntax for specifying default restart

	semantics for elements.
RestartTiming	This module defines an attribute for controlling the begin behaviour of an element that has previously begun.
SyncBehavior	This module defines syntax for specifying the runtime synchronisation behaviour among elements.
SyncBehaviorDefault	This module defines syntax for specifying default synchronisation behaviour for elements and all descendants.
SyncbaseTiming	This module defines the attribute value syntax for the begin and end attributes that allow elements to begin and end relative to each other.
SyncMaster	This module defines syntax for specifying the synchronisation master for a timeline.
TimeContainerAttributes	This module defines attributes for adding time container support to any XML language elements.
WallclockTiming	This module the attribute value syntax for the begin and end attributes that allow elements to begin and end relative to real world clock time.

Table A.1: SMIL synchronisation modules

Appendix B: Project schedule

	2002	*2003*		*2004*		*2005*		*2006*	
Research Activities	*Oct-Dec*	*Jan-June*	*July-Dec*	*Jan- June*	*July- Dec*	*Jan-June*	*July-Dec*	*Jan-June*	*July-Dec*
Literature survey	🟨	🟨	🟨						
Writing Chapter 2 'Literature Review'		🟨	🟨						
Analysis and selection of tools									
J2ME/ JSAPI analysis		🟨							
Analysis of mobile media players		🟨	🟨	🟨					
Analysis of networking tools		🟨	🟨						
Review of AI techniques & other reusable system components (e.g. CPNs, NLP)			🟨	🟨					
TeleMorph design			🟨	🟨	🟨				
Architecture implementation									
Implement AI technique for constraint analysis & modality selection (e.g. CPNs, HUGIN)				🟨	🟨				
Develop TeleMorph client application components (TeleTuras)				🟨	🟨				
Develop remaining TeleMorph server components and integrate with client				🟨	🟨	🟨			
Reiterative improvement & other units						🟨	🟨		
Integration and testing				🟨	🟨	🟨	🟨		
Improving system							🟨	🟨	
Write up PhD thesis							🟨	🟨	🟨

Table B.1: Project Schedule