Position-aware Speech-enabled Hand Held Tourist Information System

9th semester project - Aalborg University
Uffe Overgaard Koch - group 99-973

AALBORG UNIVERSITY
INTELLIGENT MULTIMEDIA
INSTITUTE OF ELECTRONIC SYSTEMS

Frederik Bajersvej 7 ▪ DK-9220 Aalborg Øst Telephone +45 96 35 80 80

TITLE: Position-aware Speech-enabled Hand Held Tourist Information System
THEME: Advanced Intelligent MultiMedia Systems
PROJECT PERIOD: 9th semester, September 1999 to January 2000
PROJECT GROUP: 973

AUTHOR:
Uffe Overgaard Koch
<uffe@data-dealeren.dk>

SUPERVISOR:
Lars Bo Larsen
<lbl@cpk.auc.dk>

Abstract

The main goal of this project has been to analyze and design an position aware speech enabled hand held tourist information system, and thru this process to learn about advanced Intelligent MultiMedia systems. By introducing position and direction awareness in a hand held computer system, the system can be used to guide a tourist around on sight seeing. The positioning is done using Differential Global Positioning System and directioning is done using a solid-state compass. A user interface consisting of the combination of speech synthesis and -recognition and the small screen on a Personal Data Assistant, is designed. The system is designed around a Dialogue Manager and dialogues for the system are designed. The system has been partly implemented, and a proposal for testing the designed system is presented.

NUMBER OF COPIES: 5
NUMBER OF PAGES: 63
END OF PROJECT: January 7th 2000

Preface

This report is the result of a project in Intelligent MultiMedia 9th semester at Aalborg University, Institute of Electronic Systems, made by Uffe Overgaard Koch. The project started September 2nd 1999 and ended January 7th 2000. The theme of this semester is: Advanced Intelligent MultiMedia Systems.

The focus of this project has been:

- To focus on theories, models and systems of advanced methods for computer display and semantic understanding of various media such as text, speech, sound and visual information.
- To learn about software platforms and architectures which are useful for integration of multiple media processing e.g. hyper media platforms, generic dialogue systems, concurrent structures.
- To investigate applications of Intelligent Multimedia for Personal Data Assistants (PDA), the Internet and mobile platforms. [E-Studyboard, 1999]

This report documents my initial thoughts of a position aware speech enabled hand held tourist information system. Existing information material aimed at tourists are analyzed, followed by a system description proposal that is analyzed and following designed. The designed system is partly implemented.

The report is set using the report style in LaTeX. Tables and figures are numerated consecutive in each chapter. Bibliographical references are indicated i square brackets by the last name of the author and the publication year. The bibliography can be found after the appendix.

I would like to address special thanks to Bo Bai for giving me an introducing me to a prerelease of his CPK Dialogue Manager v2, Bosch Telecom A/S for lending me the equipment, Sonofon for sponsoring the phone account and Thorbjørn Nielsen from Institute 4 at Aalborg University for supplying me with world coordinates.

Uffe Overgaard Koch

Contents

Chapter 1

Introduction

The purpose of this project is to investigate the possibilities introduced by including position awareness in a hand held multi modal computer system. Especially the new possibilities in a hand held system, when combining a GPS receiver and an electronic compass with a Personal Data Assistant (PDA) => position and direction awareness.

As a sample application, the needs and requirement for a tourist information system is analyzed, and a system is designed and partly implemented and tests regarding to usability have been planned. This system is thought to be handed out or rented to tourists when they arrive to a city or when they want to go sightseeing, or the system could be a download-able application to a common personal hand held device of the future the tourist already have.

In his book "Hitchhiker's Guide to the Galaxy" Douglas Adams provides Ford Perfect, one of the main characters of the book, with a special electronic travel guide. Whenever Ford Perfect needs a piece of information, he just types in a keyword and gets all the desires data. Sometimes even more.

The hand held system developed in this project will offer the tourist information about hotels, sights, restaurants, museums, bars, railway stations, airports, taxi parking places, etc. This information should contain names, descriptions and routes to the places from the users current position.

The system developed in this project will have three main functionalities/modes:

Guided tour: The tourist follows the walking directions given to him by the system, while the system is informing him about the sights as he reaches them on his tour.

Directions to one place: If the tourist wants to go to a specific place he can ask/tell the system to guide him there.

Independent exploration: The user can walk around the city and if a place or building catches his interest, he can point the hand held device at the spot and ask the system what it is.

These modes should be nestable so the user for example is on a *Guided tour* and asks for information about a building, the system enters *Independent exploration* mode, and when finished it returns to the *Guided tour* mode where it left of. Guiding the tourist back on track if he wandered of the route during his explorations. The route should continuously be updated to always have the users current position as a starting point.

An ideal system would consist of one small integrated device like the Nokia concept phones in figure 1.1.

Figure 1.1: Nokia concept phones. [Kjærgaard, 1999]

Chapter 2

Preanalysis

In this chapter the environment the system is going to be used in and the users of the system, are described and analyzed. This is done by first describing the existing tourist information material and -systems, some common components from existing material and their adaptability to the modalities of a position-aware speech-enabled hand held tourist information system. The diversity in types of tourists is described, and finally some thoughts on an online tourist information system are presented. This is all done to form some idea of how the system and the dialogue between the user and the system should be designed.

2.1 Problem area

In this section existing information provided to tourists and information systems aimed at tourists are described and the individual components of the existing material are evaluated for use in a hand held information system.

On October 19th 1999, I visited Aalborg Tourist Council (Aalborg Turist og Kongres Bureau). I asked the clerk for a set of material that are handed out to tourists when they come and ask for information about the center of Aalborg. He gave me two leaflets: "Aalborg Guide" and "Good Old Aalborg". Alternatives to these materials are web sites like: *www.aalborg-turist.dk*. Another alternative was the InfoVision system developed by AM Production Multimedia. The advantages and disadvantages of using a human tour guide are also discussed. The five different materials are described according to their media, format, information, and are rated by my subjective opinion.

2.1.1 Aalborg Guide

The Aalborg Guide is a 50 page leaflet published annually by Aalborg Tourist Council. It is financed partly by the Tourist Council and partly by commercial advertisements, these adds take up about 30-50% of the guide. The guide has a circulation of 375.000 every year,

and is printed in four languages: Danish, Swedish, English and German.

The guide contains some introductionary texts about the city of Aalborg, briefly going over the history of Aalborg and what the city is famous for. There are a number of paragraphs about sightseeing in Aalborg, how to get around and which guided tours that are available. Information about where to find libraries, cinemas, shops, parking, banks, etc. is included on list form. On the center-fold of the leaflet there is an eight page fold out map of Aalborg and its closest suburbs, with number references to an extensive list of the sights. On the last page of the leaflet there is a two page foldout map of the center of Aalborg, in a slightly smaller scale than the overview map in the center of the leaflet. [Council, 1999a]

2.1.2 Good Old Aalborg

Good Old Aalborg is a 16 page guided tour of the old buildings and places in the center of Aalborg. The guide is aimed at a tourist walking around on foot. There are two different tours in the leaflet, but they can be used as extensions of each other. The guide is well written, and is very interesting reading, spiced with small anecdotes about historical figures and places. [Olesen, 1998]

2.1.3 InfoVision

In the period from 1992 to the fall of 1998 the InfoVision tourist information system was in operation. InfoVision was a series of touch screen terminals also known as kiosks, with information about sights, shops and activities in the local area. At it's peak about 50 kiosks was in operation, spread over most of Jylland and Fyn. Kiosks was placed in shopping malls, railway stations, along motor ways and in other public areas. Half of the kiosks were owned by local media house AalborgTVR, the other half were owned by Århus Stiftstidende (newspaper), all paid for by local advertisers. Advertisers had a choice of three different types of advertisements: 1 still image, 5 still images with sound or video clips of up to 30 seconds. The kiosks were updated and maintained by AM Production Multimedia K/S.

The user could walk up to the system and select his language (Danish, English and German available), then select what kind of information he was interested in (e.g. Entertainment and Amusement parks), and then see a list of places that fit into that category. By pressing his finger on one of the icons representing the advertisers, he would be presented with their presentation/advertisement. In Aalborg Kommune a simple route planner was added, and the route description could be printed out on a small piece of paper.

The system was based on Phillips CD-I technology combined with self developed interface to the touch screen. The InfoVision project was terminated because the growing need for maintenance and the need for a technology update.[Hesselgrave, 1999]

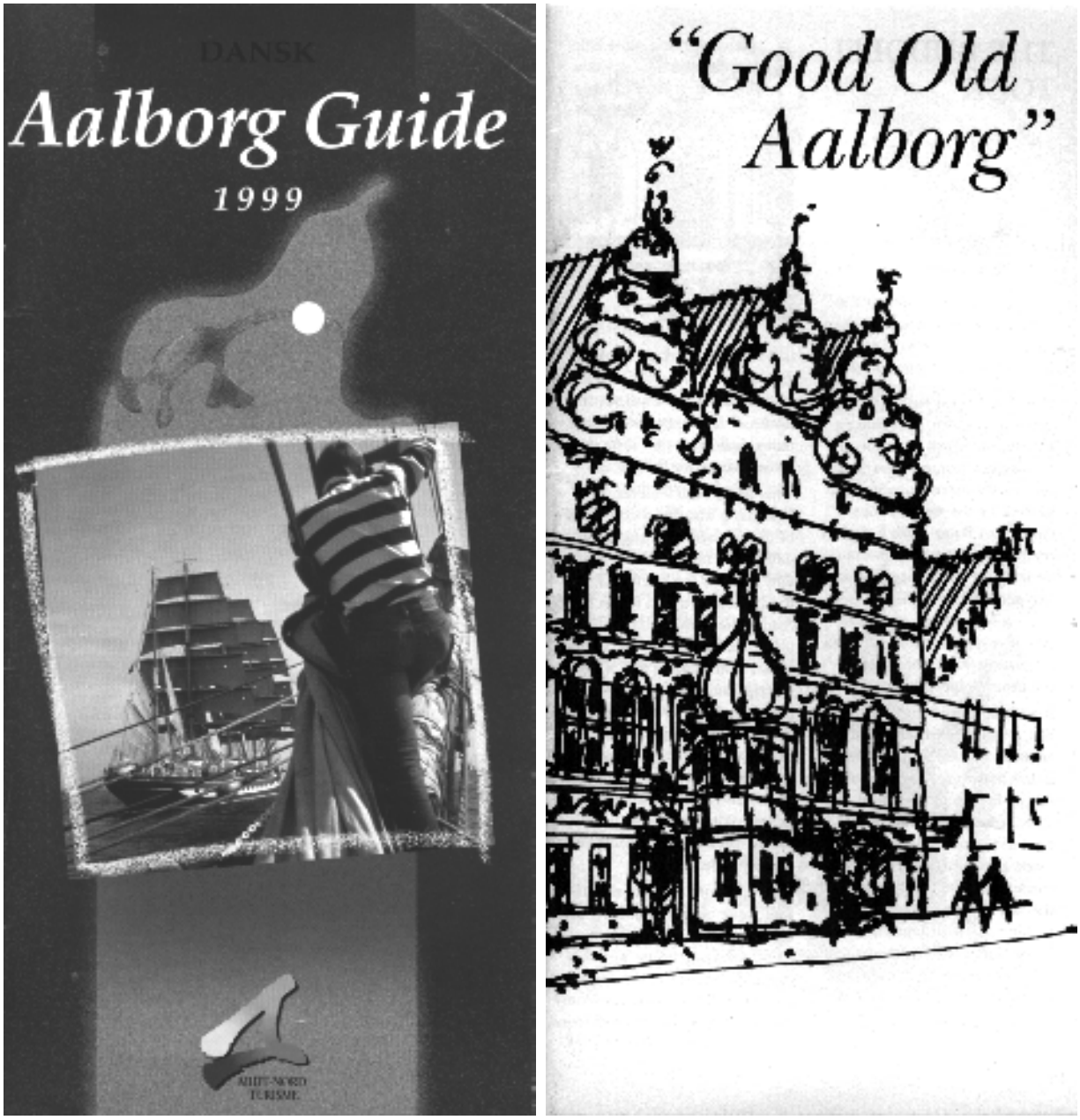

Figure 2.1: Aalborg Guide[Council, 1999a] and Good Old Aalborg[Olesen, 1998]

2.1.4 Web sites

There are an number of web sites about Aalborg but only a few directed at tourists going to or staying in Aalborg. The web site developed and maintained by *Aalborg Turist og Kongres Bureau*: `www.aalborg-turist.dk` has been picked out and is described.

The web site is organized with eight items on the main menu, illustrated using buttons with text and icons. The site has categories like: Night life, Places to sleep, Events, Tourist Services and others. The amount of information concerning individual sights is very limited, consisting of name, one line description, opening hours and a small picture. The site also includes information about tours around Aalborg and Northern Jutland, but the descriptions are very shallow. In my opinion the navigation of the site is not very intuitive and not all thru logical. [Council, 1999b]

2.1.5 Human guide

Another common way of getting to know a town or sight, is using a human guide.

This has some advantages, for instance a tourist can ask for details about a certain sight or detail. A human guide can also alter his/hers speech according to current events or the tourists interests or origin.

Human guides are in Denmark almost always guiding whole groups around at a time. This has a number of disadvantages, including that the guide often have to guide tourists from several countries at a time. This forces the guide to use different languages in turn, often loosing the attention and interest of the rest of the group while not speaking their language. Another disadvantages is that these tours are on fixed schedules, not when the tourist wants a tour. It is also hard for the tour guide to fulfill the needs and wishes of the individual tourist regarding level of detail, an average of the group must apply.

This sub analysis was made from educated guesses made by the author, a better way would be to go on a guided tour and observe the guide and tourists, maybe supplemented by a number of interviews with the guide and some tourists.

2.1.6 Information components and their cross-media qualities

A number of information components common to two or more of the mentioned material, have been identified, and their quality on a speech enabled PDA system are discussed:

1. **Activity calendars:** An overview of concerts, fairs, shows, movies etc. in the near future and returning annual events. Since most PDA's have builtin calendars, event calendars are easy to implement on PDA's. But PDA calendars are often used as a day-to-day calendar, with limited possibilities of overviewing larger periods of time.

2. **Advertisements:** Local merchants advertising their shops or products to the tourists. Could be used as a way of sponsoring a tourist information system. The commercials

would have to be intrusive to be viewed, or the shops could be in the form of sights the user can select to go to, and the description of the "sight" could hold the actual advertisement.

3. **Area information:** Information about the whole area in subject. Pictures, history, facts, practical information. This too could be presented in the form of sight description screens that the user can select to view when using the system.

4. **Maps:** Maps of the area of different types:

   a. **geographical 2D maps** with sight names placed at the place of the sight. Maps can be converted to black and white images that can be viewed on the PDA screen. PDA's have limited screen sizes and resolutions, so maps have to be very coarse or shown only in part at a time. But combining a PDA with a positioning device, the system can automatically update the part of the map shown as the user moves around. Also the possibility of showing only important information to the user, will improve the experience using the system. Maps can hold more information and be read more easily if a PDA with color screen was used.

   b. **perspective 3D maps** drawn from above, with the sights drawn as 3D images larger than their actual scale. 3D maps with detailed drawings or images of sights, is not very fit to be shown on a lo-res PDA screen.

5. **Sight descriptions:** Descriptions of individual sights, including name, age, picture, usage, other related sights, historical significance and related anecdotes. Textual information can easily be shown on a PDA screen, and descriptions in form of longer sentences can be spoken to the user so he has his eyes free to look at the sight. Detailed pictures are not fit to be viewed on a PDA screen.

2.2 Tourist / User types

A large number of tourist types exist:

Individuals can be pursuing a wish to study a town or some sights, just killing time, or have any other reason.

Couples can be young or old, wanting just to enjoy their vacation or to learn a lot, etc.

Families with or without young children, long tours or short tours.

Young people maybe want more visual and action type of stimulation than

Older people that are maybe more interested in history or architecture

Groups of students, school classes, interest groups or charter tourists all have different wishes and needs.

With so many different user types it is obvious that there is a need for a high level of user influence on the sights to be visited, and the way they are presented, and thereby interaction in the system. But on the contrary users may also want just to be guided around town, and not to be so active in the planning process.

2.3 Online system

The value of a system like the one presented in this project could be enhanced by adding online information services. One technology that spring into mind is the Wireless Application Protocol (WAP). This is a protocol designed by a consortium named WAP Forum, founded in June 1997 by information technology companies Ericsson, Nokia, Motorola and Phone.com. Since then more than 100 companies have joined WAP Forum. WAP has become the De facto standard for delivery and presentation of wireless information and information services on mobile phones and other wireless devices, e.g. wrist watches.

An example of what WAP could add to the system at hand could be the user passing a the Students house in the center of Aalborg, and then asking the system what is going on there this evening. Then the system could access the online calendar for the Students house, and present the program to the user via the WAP protocol.

Another example could be the system warning the user that the weather forecast predicts rain, when the user asks to be guided to a certain place, and the estimated duration of the walk correlates with the forecast for rain. In thread with this example the system could guide the user via alternative routes when routes are temporarily closed by road maintenance or long cues, etc.

The user could additionally use WAP to: pay for museum entry fees, book tickets for concerts, check travel schedules, and many more applications.[Danielsen, 1999]

All these features could be implemented into the system by using WAP, and provided the needed services are available. WAP can provide a common infrastructure for wireless information retrieval. WAP will not be included in the project, but is mentioned to name one way the system could be extended in the future.

Having an online system like the described, raises possibilities to track and record the whereabouts of the user and the use of the system. This information could be used as a theft protection as well as to improve the system and the information supplied with it. This could be an ever ongoing invisible usability and quality test of the system and the information material contained.

Chapter 3

System analysis

In this chapter the entire system will be presented, and the individual components and parts will be analyzed. In figure 3.1 the block structure of the proposed system is presented. It is divided into four parts:

Model defining the problem area and supplying an object model for the system being a mirror of the real world problem area.

Functions needed by the user- and system interfaces to manipulate the model.

User interface together with the system interface are used to interact with the surroundings of the system. Specifically the user interface is used to interact with the user. This is typically thru a WIMP interface, but can also, as in this case, be speech technology and hand held devices.

System interface components enables the system to communicate with external systems or hardware. [Mathiassen et al., 1993]

3.1 Model

The object of the model is to depict the real world, in a computer based object model. This model will be based on the knowledge the system designer has of the problem domain.

3.1.1 Domain Knowledge

The knowledge of the system consist of a set of sights and roads and their absolute positions in the real world. This type of information related to places and roads etc. is often stored using Geographical Information Systems (GIS). GIS maps are divided in 5 major types: [Nordjysk-Informatik-Forum, 1998]

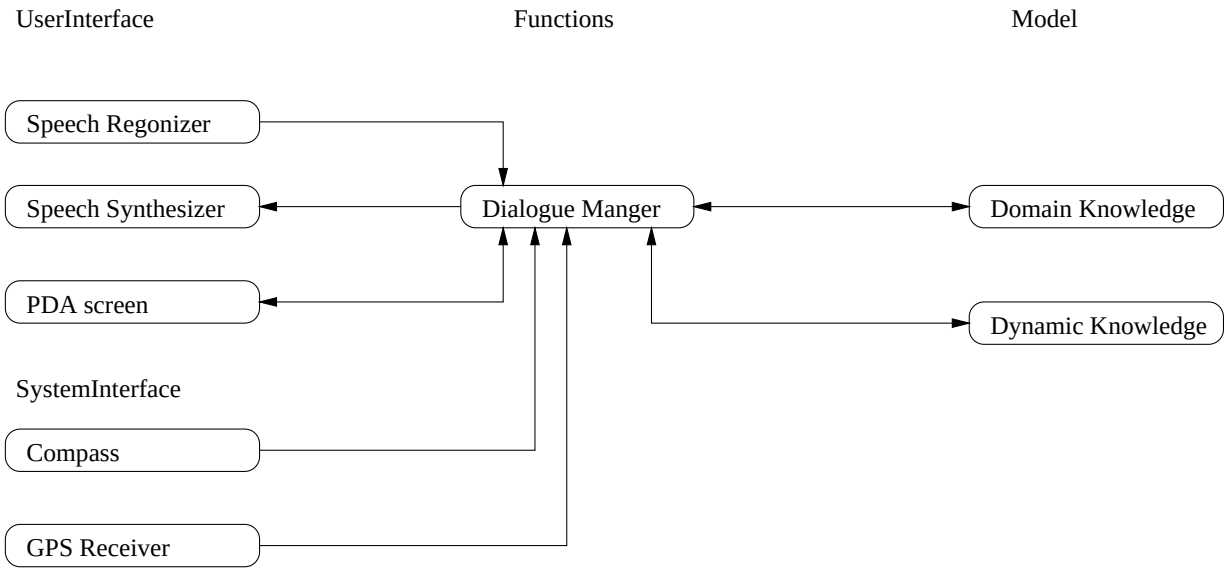

Figure 3.1: Architecture of the system

thematic maps: eg. population density

topological maps: eg. train maps, relative not absolute

topographical maps: eg. road maps, 4cm cards

technical maps: eg. road lights, drains, electric wiring

photo "maps": air or satellite photos

The requirements of the user and of the system are different on a tourist information system.

The user needs to be presented with a map, with a close and easy mapping to the real world. In this perspective a photo would be the best, but aerial photos do not have very high resolutions, so a topographical map would be easier to understand. This would be referred to as the user map.

The system need some knowledge about where the sights are, and how the user can move between them. This could be done using a topological map, with nodes in all sights and road junctions. This map would be referred to as the system map. This approach implies that graph theory can be used for calculating routes between sights and way points.

If information about the real world location of each system map node is included, a mapping between the user map and the system map can be found. By overlaying the sight nodes from the system map on the image of the user map, both the requirements of the user and the system can be met.

The knowledge about each sight must be stored in a database. At least four attributes for each sight must be stored:

- Name
- Type
- Location
- Description

The description must be stored in different level of detail, so that users that are only a little interested in a particular sight can get a short description of a sight, but the system should at the same time be able to tell more interested users a lot about a sight. This could be done by writing the short descriptive text about each sight like the teasers at the top of newspaper articles, summing up the story. The more elaborate description could then take on more the form of the article body. This gives the user the opportunity to ask for more information about a sight that catches his interest after he has been presented with the short version of the description.

3.1.2 Dynamic Knowledge

The knowledge and information the system acquires during operation, is stored and processed in the Dynamic Knowledge module. This is primarily information about the progress of the dialogue between the user and the system, and is used as a semantic recording mechanism.

This module is required to keep track of all interaction with the user input and output, and to keep track and record of all decisions made by the system, on basis of static knowledge combined with user input.

This module also continuously controls whether the user is on the right way, or if he is diverting from the route. If he diverts too much from the route he is notified by a spoken warning, but only once, after this the system calculates a new route to the original destination, from where the user is now.

3.2 Functions

In this section the requirement to the design of the functions required for communicating reasonably between user- and system interfaces and the model of the real world, will be described.

3.2.1 Dialogue Manager

A dialogue manager is a mechanism that keeps track of the structure and the state of a dialogue. Dialogue managers have different architectures depending on the nature of the dialogue. The dialogue manager can be independent of the modalities used in the interaction, but often they are closely related to one or two modalities.

The process of designing a dialogue consists of an initial analysis of the application area and a recursive process of prototyping, implementing and testing.

The analysis can be based on interviews with future users of the system, domain experts, domain literature and other heuristic studies. This analysis should form the basis of developing the dialogue. Second step in the analysis process is often done using scenario based analysis, where a number of sample dialogues are constructed, either by Wizard of Oz studies or "educated guessing" by the developer. This last approach is used in this project, and the scenario based analysis can be found in appendix A.

The sample dialogue in appendix A shows that the system should be able to process deictic references ("What is *that*?"), which is very characteristic and important to multi modal systems. It is also necessary for the system to handle spatial relation references, so sentences like "You are now *in front* of Jens Bangs stenhus" uttered by the system, can be constructed and uttered. As seen in the sample dialogue the system should be able to resolve ambiguities "*Which* church do you mean?". The Dynamic Knowledge is used by the system to be able to understand user references to previously mentioned sights/places without having to give complete reference every time *it* is referred. [Brøndsted et al., 1998, p.16]

The analysis is followed by a design process, where the actual sentences going in and out of the system are defined. In this context sentences are actual sentences uttered or recognized by the system and other user interaction with the system, in this case actions to and from the PDA screen. These sentences are also defined in the appendix.

The system is communicating with the user occupying a number of the users senses. The user may wish to use the system without the PDA screen or without the headset with headphones and microphone, still getting information about the sights and route followed. The design of the dialogue should bear this in mind.

Another important aspect to consider is that users may not allow the device to be dominating or intrusive, so the system have to allow the user to control whether speech is allowed at the time.

3.3 User Interface

In this section both the physical and the logical analysis of the user interface modalities will be described.

3.3.1 Speech Recognition

Speech recognition is in this project defined as the process of extracting some meaning or semantic of a continuous sampled human speech signal. In other contexts speech recognition also includes word spotters, speech recognizers trying to match individual words.

Speech recognition systems can be classified in two categories:

Dictate systems: e.g. Dragon Dictate, IBM ViaVoice

Small/Medium Vocabulary systems: e.g. GrapHvite, HTK, IBM ViaVoice

IBM ViaVoice is listed in both categories because it can be used both as a dictate system and as a rule grammar based system.

Other features distinguishing recognizers are:

Speaker dependency Systems can either be speaker dependent or independent. Speaker independent recognition is much more difficult than speaker dependent recognition. Speaker dependent systems must adapt phoneme models to individual users, by training of the models.

Languages Speech recognition systems come in a variety of languages: English, French, German, Spanish, etc. No commercial recognizers support Danish yet.

All types of speech recognition systems have a number of common feats:

- Vulnerable to noise. This can be a problem because the system is intended to be used outdoors near traffic and other noise.
- Distinct pronunciation necessary. Because users are tourists with all types of backgrounds, common users are not used to using speech recognizers, and may thereby not speak clearly enough.
- No major variation in speech rate or intonation allowed. The user has to speak in a "monotonous" voice.

For this project a speech recognition technology that support recognition of continuous English speech based on rule grammars, that is fairly noise insensitive, is needed.

3.3.2 Speech Synthesis

Speech output is used in the system to guide users, to describe sights and to resolve ambiguities in the dialogue. This can be achieved by generating pure synthetic speech by applying text-to-speech (TTS) synthesis to a text string. Another method is to use pre recorded human speech. The speech is recorded as single words or short phrases and then

concatenated into sentences and played back to the user. This method will under some circumstances sound more natural than synthetic speech but has the problem of being very inflexible. [Brøndsted et al., 1998, p. 58]

The amount of texts that are to be spoken by the system is rather large and very dynamic, so a TTS solution is necessary. For this project a speech synthesis technology that supports on-the-fly TTS generation and playing is needed.

3.3.3 PDA's

In this section the definition of a Personal Data Assistant(PDA) used in this project, will be given, and the differences in user interfaces that set this type of computers apart from desktop computers.

Figure 3.2: HP Jordana 430se and Compaq Aero 1500.[Ølholm, 1999a]

In 1992 the Apple Vice President John Sculley used the term PDA the first time to announce the Apple Newton PDA. PDA's are portable, in combination with a wireless modem you can access information services from where you want and need it. PDA's can operate in any position. No matter if you are standing, sitting, walking or lying. In PDA's the computer technology is invisible to the user so they might not be as anxious in doing something wrong as they might be using a conventional desktop computer. A pen (or stylus) is more natural (everyone is used to to write with a pen) and it is easier to operate an application with it compared to a mouse. The proper use of a mouse requires some training, although this is now considered common knowledge. [Gessler and Kotulla, 1995]

PDA's are multipurpose programmable hand held computers, not bigger than approximately 15cm at any edge. Programmable calculators like HP48 and TI89 are not PDA's, their primary task are to be calculators. Small databanks from companies like Casio for

storing addresses and phone numbers are not PDA's either. Mobile phones of today even with data access like Nokia 7110 with WAP connectivity, are not advanced enough to be considered PDA's. All PDA's on the market are designed primarily to be organizers (note pad, address book, calendar, etc.), but can in fact be considered small multi purpose computers.

PDA's are in this definition computers like:

- 3com Palm Pilot
- Apple Newton
- Casio Cassiopeia
- Compaq Aero
- Handspring Visor
- HP Jordana
- Psion 5
- ...and many other

Figure 3.3: Casio Cassiopeia E-105[Computer, 1999] and Handspring Visor[Ølholm, 1999b].

These PDA's are build with a large variety of CPU's but all running one of a few select operating systems(OS):

PalmOS: Palm Pilots, Handspring Visor

MacOS for Newton: Apple Newton

Windows CE: HP Jordana, Compaq Aero, Casio Cassiopeia

EPOC: Psion 5

Of these OS's three are still under development and spreading rapidly: PalmOS, Windows CE and EPOC. PalmOS devices currently hold approximately 63% of the world market for PDA's (March 1999). There has been sold more than 3.2 million Palm Pilots(March 1999) world wide.[Pedersen, 1999]

All these OS's have builtin widget libraries like desktop OS's, with buttons, text boxes, windows, sliders, etc. All these widgets are designed for easy viewing on small screens with few colors. Most of the current PDA's have monochrome screens, but new models are coming out with 16 or more colors. Most screens have approximately the same resolution as desktop monitors, e.g. 3com Palm Pilot has a 160 x 160 pixel screen of 5.4 x 5.4 cm, which yields a resolution of 75.2 DPI (dots-per-inch). A typical desktop monitor have resolution in the range from 72-85 DPI.

All of the PDA's have a few buttons on the case, one used as power switch, the rest used as shortcuts to most used functions or programs.

Most of the mentioned PDA's uses pens or styluses as input device, using some or all of the screen as input area. This maximizes the writing area on the small devices. This is a very limited input device only allowing the user to tap or use strokes as signals to the PDA. The PDA's widely uses menus and icons, allowing the user to tap the selected action. Using strokes the user can input letters either by using his own handwriting style or by adapting to a alphabet defined by the PDA. With a bit of training this can be an acceptable form of inputting short text messages, like appointments in calendar programs or notes to one self. Not quite as fast as typing on a keyboard, but acceptable for short messages.

When designing a user interface for a PDA the designer has to make special considerations compared to designers of desktop computer applications. Screens are very small so large amounts of information have to be presented in smaller bits and the user have to scroll a lot more, still not loosing the overview of the information. A common misconception is that the user thinks that he is seeing all of the information, when in fact he may only be viewing just a corner of the information. The small screen size can also be an advantage, because the user can overview the full screen without having to change the focus of his eyes or to turn his head. The designer can only use one or at the most two windows with very limited amount of information in them, at the time. The user only has the stylus for interacting with the program, so complex actions have to be made easily accessible. This might be done using modes, but modes are known to confuse novice users.

For this project a 3com Palm V is supplied by Bosch Telecom A/S so this will be used for interacting with the user together with speech synthesis and -recognition.

3.4 System Interface

This section holds the description of requirements to interfaces to external systems; compass & positioning system.

3.4.1 Compass

In this project a compass is used to determine the heading of the user. Because the compass have to work, even if the user is not holding it still, a solid-state compass is needed. Solid-state compasses can be constructed in a number of ways, but all types are based on measuring the horizontal component of the earths magnetic field. The *Valvo KMZ10A Magnetic Field Sensor* is sufficiently sensitive to measure the magnetic field down to a precision of +/- 5 degrees, which seems sufficient for this project. The sensor utilizes the magneto resistive effect: resistance from thin film resistors varies with external magnetic fields and can be pre-magnetized so that the field to be measured increases or decreases the resistance. [Wellhausen, 1989]

3.4.2 Positioning systems

There are a number of different technologies that can provide different degrees of precision in positioning. These descriptions will form the basis of the choice of technology, in the design phase of the report.

GPS

GPS is a satellite based navigation system, built and run by the American Department of Defense (DoD). The GPS system consists of a least 24 satellites covering the earth, arranged on 6 equally spaced orbit planes at an altitude of 20.000km above the earth, with 4 in each. The satellites transmit signals, that a hand held GPS receiver can use to calculate it's current position.

The system offers precisions from around 100 meter down to centimeters, depending on equipment and measuring methods. There are a number of different receivers on the market, and the system have been fully operational since 1982. [Borre, 1995]

Due to the fact that the DoD do not want to give their enemies a perfect positioning system for use in terrorist or military action directed towards the US, a distortion scheme is applied to the GPS system. This system is named Selective Availability (SA), and is a random signal added to the GPS signal transmitted by the satellites. The SA worsens the GPS positioning capabilities to an average precision of 20-40 meters. [Nielsen, 1999a]

Other major error sources are:

Ionosphere-delay: The number of electrons in the Ionosphere creates a variant delay, in daytime 20-30 m, and at night 2-6 m.

Troposphere-delay In the lower part of the Atmosphere an error of up to 30 m for satellites close to the horizon is causing a delay. This error is almost constant and is easily modeled.

Satellite orbit-error: The difference between the true orbit of the satellite, and the calculated orbit position. This error is usually less than 3 m. [KMS, 1999]

DGPS

One way around the SA and other errors, is to use Differential GPS (DGPS). By placing a GPS receiver on a known spot and keeping it there, the difference between the current GPS signal and the actual position can be found. This difference is often referred to as the error or the error signal. By connecting to a DGPS service close (less than 600km) to the current position, the error can be subtracted from the GPS signal, and a far better position can be found. Eg. down to 2 m. [Nielsen, 1999a] [KMS, 1999]

> Policy Guidelines:
>
> 2.It is our intention to discontinue the use of GPS Selective Availability (SA) within a decade in a manner that allows adequate time and resources for our military forces to prepare fully for operations without SA. To support such a decision, affected departments and agencies will submit recommendations in accordance with the reporting requirements outlined in this policy.
>
> [http://www2.whitehouse.gov/WH/EOP/OSTP/html/gps-factsheet.html]

In Denmark the public institution "Kort og Matrikelstyrelsen" offers two ways of receiving a DGPS signal: Spot-FM & NAV-DK. Spot-FM uses a digital radio channel named FM-RDS (often used in car radios) and NAV-DK uses a GSM data service. The price (autumn 1997) of Spot-FM is 2500DKR for e receiver and 3500DKR per year for the service. The price for NAV-DK is 8000DKR for a data interface and 2,35DKR per minute. [KMS, 1999]

There are plans of making a free of charge public DGPS station at Aalborg University that transmits a DGPS signal via the Internet.

GLONASS

GLONASS (GLObal NAvigation Satellite System) is the Russian pendant to the GPS system. The system is very similar to the GPS system, but has no SA scheme, and is thereby more precise. The system consists of 24 satellites in fixed polar orbit, like the GPS system, but now only 15 are still in use. It is hard to get dedicated GLONASS receivers, the receivers are most commonly cased together with GPS receivers. [Walsh and Daly, 1998]

GSM

GSM positioning is still very new in the market, but there are already a number of products available on the market, that makes it possible to use mobile telephone signals for positioning. The system works by triangulating the signals from the cellular phone antennas in a central computer, and thereby estimating the position of the user. The system has right now a accuracy of 100-300 meters, depending on the area. This technology is still very new, and it will probably increase in precision in the future. The question of privacy is compromised though, with this method. This issue is to be considered individually in each application.

Requirements and choice of technology

For this project it is necessary to distinguish the users position with a precision of few meters, because some sights and way points lie very closely. The only positioning technology that fulfills this requirement is Differential GPS (DGPS).

3.4.3 Geodatums

There are a large number (>100) of different coordinate systems in use around the world, these systems are referred to as geodatums or just datums. When a position is given the datum is equally important as the longitude, latitude and altitude, but is often neglected.

In Denmark the datum used for topographical maps is ED50, but in many technical maps a datum named System 34 is used. GPS has its own datum named World Geodetic System 1984 (WGS84). In Denmark a large number of WGS84-coordinates has been determined with great accuracy. This is also known as the realization of WGS84 in Denmark, this is the actual manifestation of this datum.

In each datum a number of projections can be used to map the three dimensional surface to a two dimensional map. A common projection used is UTM. This means that a map can be in datum ED50 and drawn in the UTM projection, and a GPS-receiver can deliver UTM coordinates in datum WGS84. If these GPS coordinates are drawn uncritically on a map with another datum, an error of several hundreds meters occurs.

This shows that a conversion between map datums and projections and GPS datums is crucial.[KMS, 1999]

Chapter 4

System design

In this chapter the design of the system, and the considerations connected with this, will be described. The goal of the design process is to reach a level of detail in description of the system, so that implementation is straight forward and trivial.

4.1 Technical choices

In this section the hard- and software platform is described and argumented.

4.1.1 Hardware

For this project Bosch Telecom A/S was kind enough to lend me and Torben Pedersen, who is working on a similar project, some hardware:

- Dell Latitude notebook PC
- Garmin GPS 25 LP - GPS Receiver
- Bosch Telecom 809 cellular phone
- 3com Palm V - Palm Pilot

The notebook computer has a 233MHz Pentium II processor and 64Mb RAM.

4.1.2 Software

In this section the choices of operating system, development language and external software packages are described, both for Palm Pilot and notebook subsystems.

Palm Pilot

Initially the intention was to run the system on the notebook and use the Palm Pilot as graphical user interface. Two Java-like virtual machines was investigated:

- Waba VM:

  Supports: GUI, Serial communication

  Does not support: Threads [Wabasoft, 1999]

- KVM:

  Supports: GUI, Threads

  Does not support: Serial communication [Sun-Microsystems, 1999b]

Since both serial communication and threads are needed, none of the current alternatives for running Java on PalmOS is applicable for this system. Thus after investigating Java (and Java-like) development languages for the PalmOS platform, the GUI will be designed for the notebook display.

Notebook

The notebook is running Microsoft Windows98, and JDK 1.2.2 was used as the all purpose software development environment. Long into the development process RedHat Linux 6.1 was used as the development operating system (OS), but version problems between ViaVoice and Speech for Java forced a change of OS.

For speech-synthesis and -recognition IBM ViaVoice was used. This was chosen for a number of reasons.

- Availability on the initial development OS: Linux.
- Availability on the final development OS: Windows 98.
- Java interface: IBM AlphaWorks "Speech for Java".
- Commercial technology, hence robust technology.

The "Speech for Java" interface is an implementation of the Java Speech Extension API, defined by JavaSoft. This will in the future give developers of speech applications a common API for using a number of speech engines. Currently only a small number of speech-synthesizers and -recognizers have implementations of the Java Speech API. [Sun-Microsystems, 1999a]

For this project the ViaVoice speech-synthesizer and -recognizer are used as black box technologies. This means that their internal structure is not investigated very deeply, and focus is on the application of the technology in a multi modal system development context.

For storing and retrieving structured data a MySQL database is used. MySQL is available for a large number of OS's including Linux and Win98. It is a general SQL database, and is supported by a JDBC driver for connecting Java applications to MySQL databases. [AB, 1999]

4.2 Dialogue design

In figure 4.1, the main dialogue graphical description is shown. For graphical description of dialogues the "Dialogue Description Language" from [Bækgaard, 1996] is used.

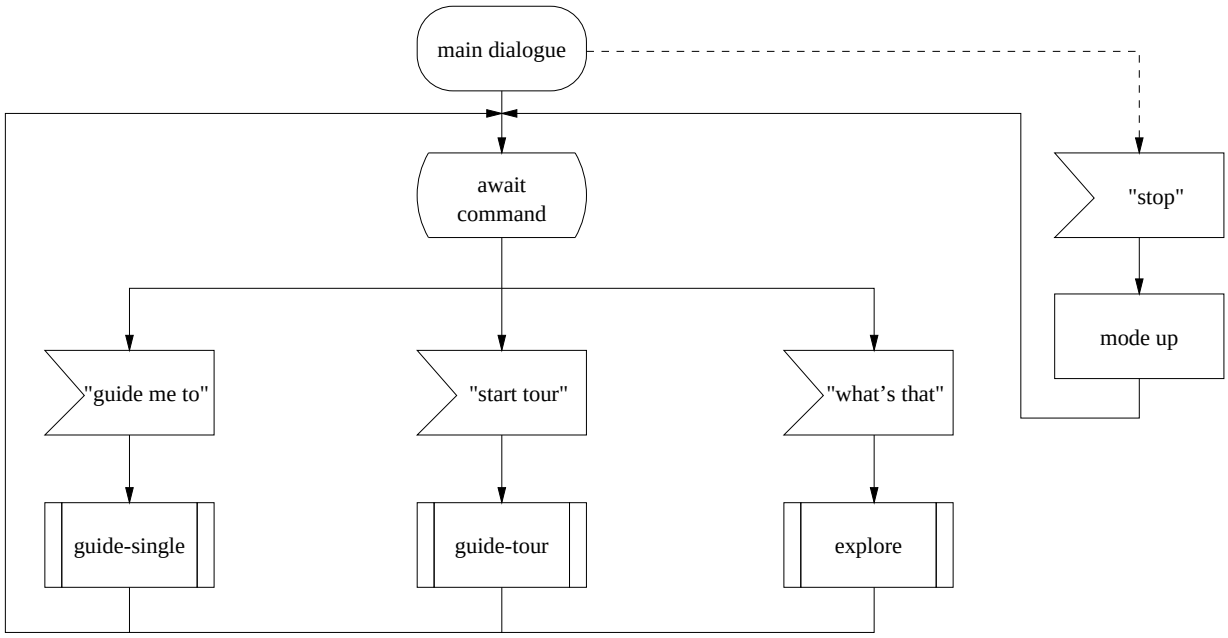

Figure 4.1: Main dialogue

The sub dialogues for `guide-single`, `guide-tour`, `explore` and `resolve-ambiguities` are found in the following figures.

4.3 Dialogue Manager

The CPK Dialogue Engine v2 from Bo Bai is used.

The dialogue manager is the heart of the system. The instance of the dialogue manager is the class with the `main` method called to start the system. The dialogue is implemented in the `dk.datadealeren.imm9.dialogue.Dialogue` class, that extends the `dk.auc.cpk.engine2.test.TestDialogue` class that supplies a number of methods for constructing, loading and saving dialogues.

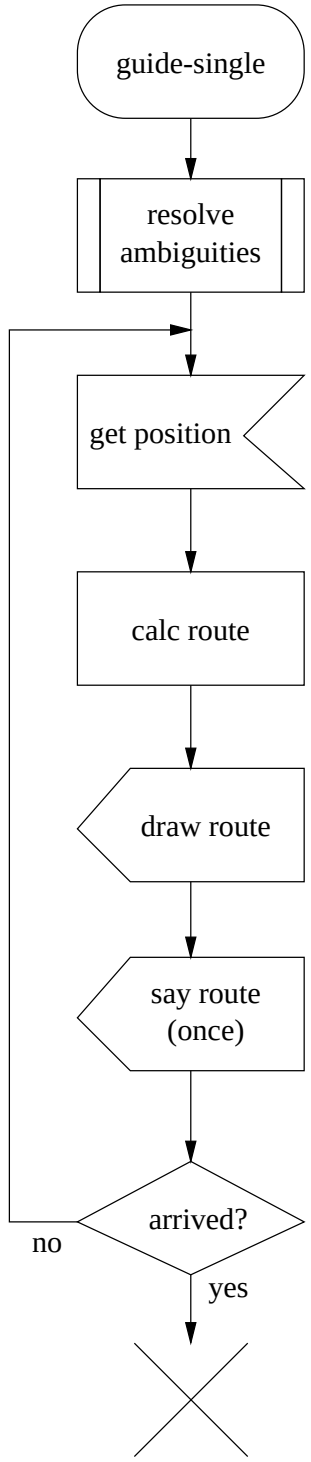

Figure 4.2: `guide-single` dialogue

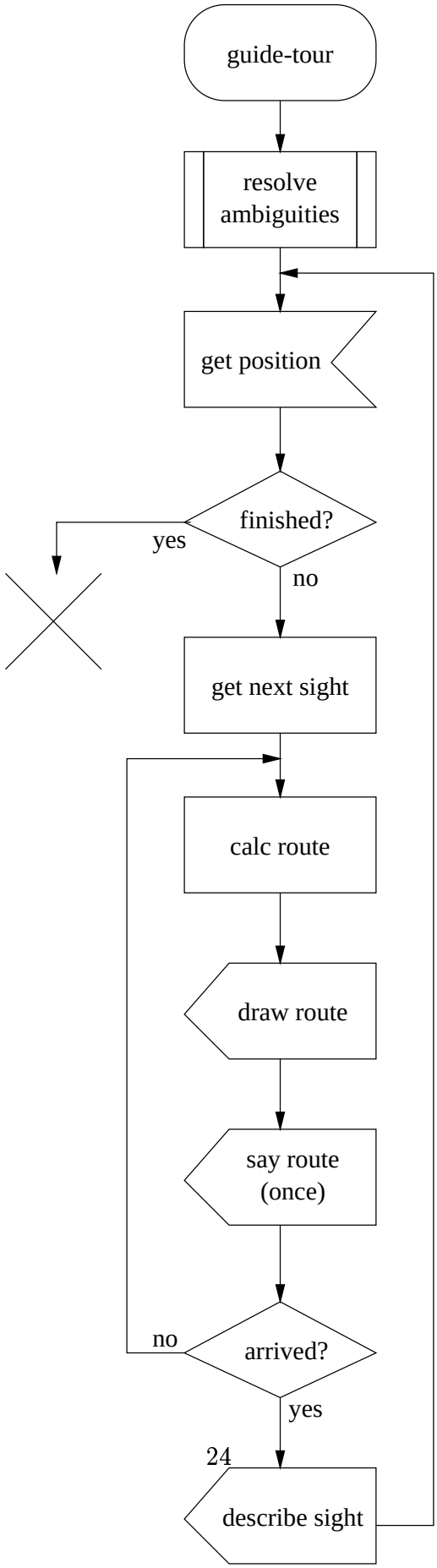

Figure 4.3: `guide-tour` dialogue

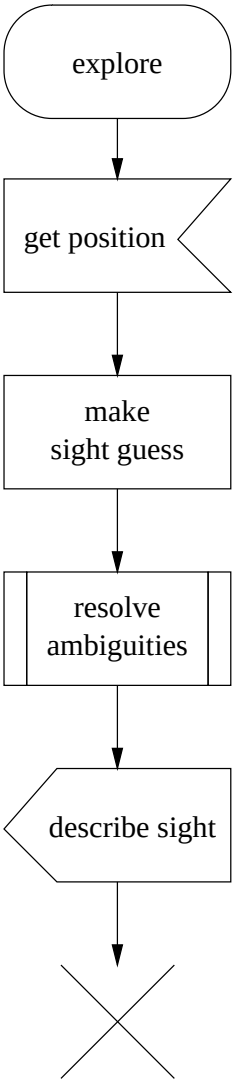

Figure 4.4: `explore` dialogue

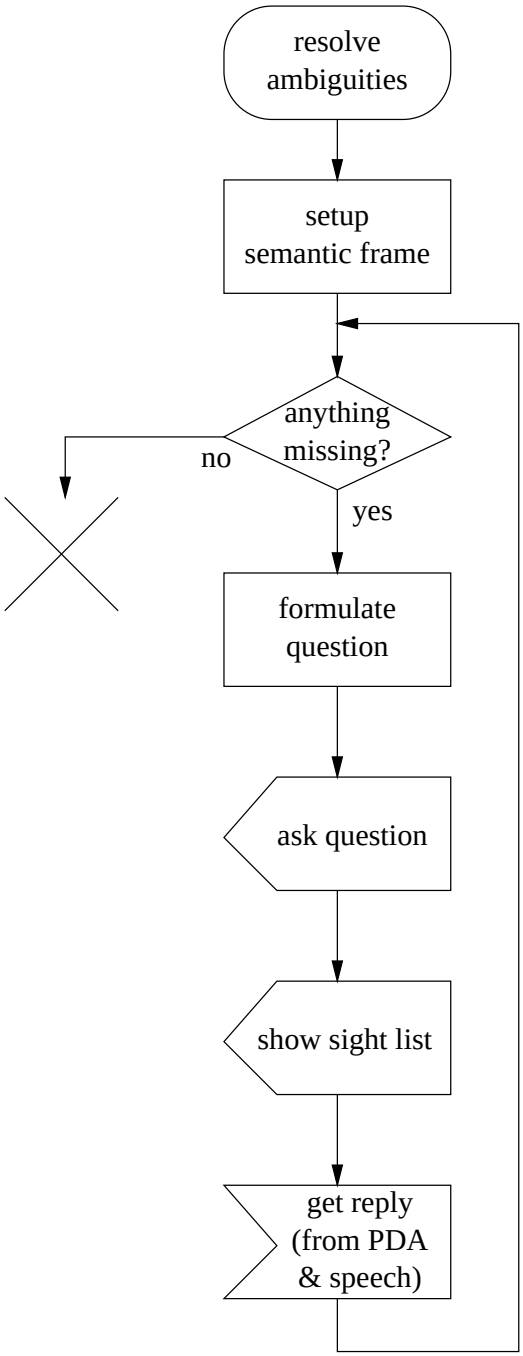

Figure 4.5: `resolve-ambiguities` dialogue

The dialogues managed by the dialogue manager consists of nodes and links. To enable the developer to reuse code for complex nodes, a command concept is included in the dialogue manager. These commands can be developed as very general modules in a dialogue system. The dialogue manager uses Java Reflection to investigate commands for `getX` and `setX` methods. All commands should implement the `dk.auc.cpk.engine2.Command` interface. This interface only holds an `execute` method, that is called by the dialogue manager when the node in the dialogue is reached. Before the `execute` method is called all `setX` methods are called to set instance variables needed by the command. When the execution of the `execute` method is finished, all `getX` methods are called to return results of the command/node to the dialogue manager.

Currently the dialogue manager is still under development, so no graphical dialogue development tools are available. Instead the dialogues has to be hand-coded.

4.3.1 Devices

A number of device handler classes are developed for handling contact with and setup of devices used in the system. These devices are:

GPSReceiver is used to initialize the serial connection to the GPS receiver. This class supplies, among others, a method that can be called to return the current position.

PDAMap is used to emulate a Palm Pilot screen, displaying the map of the area, the current user position and the current route.

PDAText is used to show lists of choices and information describing sights.

SpeechRecognizer allocates and initializes speech recognizer using the Java Speech API. This class does not supply the recognizer with a grammar, this is done when the recognition commands are used.

SpeechSynthesizer is allocated and initialized.

4.3.2 Commands

The commands developed for the dialogue are:

DescribeRoute uses the `Speak` command to speak out the route created by the `RoutePlanner` command, as well as showing the route on the map using the `PDAMap` command and device.

DescribeSight is used to describe a sight to the user by showing information about the sight on the PDA screen and by describing the sight to the user by using the speech synthesizer. When the node is reached either a sight ID number or the name of the

sight is known. This information is used to select the text to be spoken and shown from the `sight` table in the `tourist` database.

Positioner communicates with the `GPSReceiver` and finds a trustworthy position, and returns this.

PDAList is used to setup a list of sight names on the PDA screen, when the system is in doubt of which sight the user is referring to.

PDAMap shows the map, and takes a route as argument, and draws this on top of the map, along with the current position.

RoutePlanner is an implementation of Dijkstra's shortest path algorithm.

Speak is used when the system have to speak. It uses the `SpeechSynthesizer` device, and is called using a string of text. This command is called both directly by the system, and by other more complex commands.

SpeechListen is used when the system expects spoken input only. It is called with a Java Speech Rulegrammar in the Java Speech Grammar Format (JSGF). It returns the tag string associated with the rule fired. Other parameters this command should accept includes timeout and synchronization information. This command must accept both static and dynamic (run-time generated) grammars.

SpeechAndPDAListListen is similar to the `SpeechListen` command, but also "listens" for input from the `PDAText` device. This command also holds the priority and chooses between simultaneous or diverting input.

Switch is a general switch command. It takes an array of `Objects` as cases and another `Object` for value. It returns a `Branch` object that shows the dialogue manager which branch to follow.

Tester holds a number of helper methods for managing the Palm Pilot mockup interface and for debugging of the system.

4.4 Dynamic Knowledge

The dynamic knowledge acquired by the system will be handled using a semantic frames representation.

This module is inspired by the Blackboard module of the CHAMELEON system, but is not placed as central module as the Blackboard in CHAMELEON. The reason for not using the Dynamic Knowledge as a central module, is the wish to investigate the mutual roles between the Dialogue Manager and Dynamic Knowledge. These modules are very closely coupled, and both essential in most multi modal systems.

The semantic frames will be structured like the frames in the CHAMELEON system, and are grouped in three categories:

1. input
2. output
3. integration

Input frames take the general form:

```
[MODULE
INPUT: input
INTENTION: intention-type
TIME: timestamp]
```

`MODULE` is the name of the module the frame originates from.

`INPUT` can be *UTTERANCE, POSITION, DIRECTION, SELECTION, TIME*, from respectively the Speech recognizer, GPS, Compass, PDA screen and system clock.

`input` is the actual input received from the module.

`intention-type`s are defined in the dialogue design.

`timestamp`s are included for logging and debugging reasons.

Output frames take a similar form:

```
[MODULE
INTENTION: intention-type
OUTPUT: output
TIME: timestamp]
```

`OUTPUT` is either *UTTERANCE, PDA-MENU, PDA-INFO* or *PDA-MAP*, and `output` is the output directed to the appropriate module.

Integration frames are non-input/output frames, and take the form:

```
[MODULE
INTENTION: intention-type
NODE: nodeid
OUTPUT: output
TIME: timestamp]
```

Integration frames can be used to: describe endophoric references, describe routes, resolve ambiguities, etc. [Dalsgaard and McKevitt, 1998]

4.5 Speech synthesis

The speech synthesis is in this project used as a black box technology. It is required of the speech synthesizer that it takes a text string as input and performs the text-to-speech conversion and sound playing. ViaVoice does this. The speech synthesizer should take annotated texts including simple emphasis, stress and intonation commands. ViaVoice can do that, as well as voice selection information.

4.6 Speech recognition

The ViaVoice speech recognition engine is language-neutral and data-driven. The speech engine has a rather complex task to handle, that of taking the raw audio input and translating it to recognized text that an application understands. Figure 4.6 depicts the logical architecture of the ViaVoice speech recognition system.

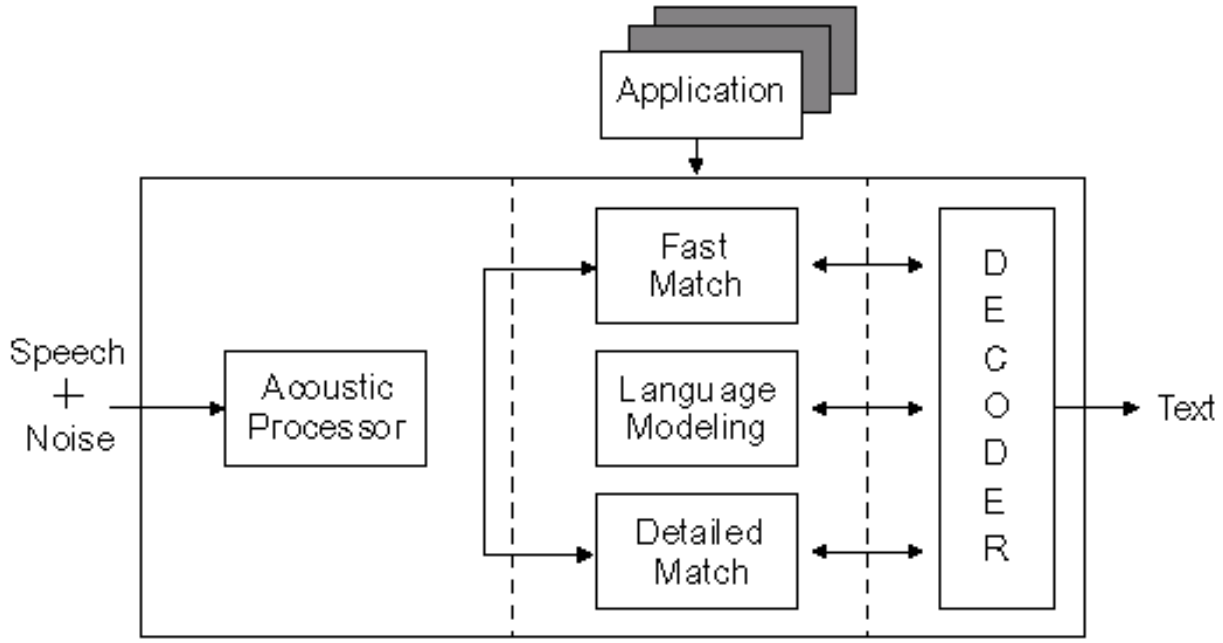

Figure 4.6: ViaVoice Speech Engine Architecture[IBM, 1999]

The audio input module encapsulates the methods used by the engine to retrieve the audio input stream. The engine retrieves its audio input from the microphone input device in the system. The acoustic processor takes raw audio data and converts it to the appropriate format for use. The acoustic processor consists of two components: the signal processor and the labeler. In the ViaVoice engine, audio input picked up by the microphone is analyzed by the signal processor. This raw audio data is captured at 22 kHz by default, but 11 kHz

and 8 kHz sampling is also supported. It contains both speech data and background noise. [IBM, 1999]

The microphone in the system can be placed either on the device it self or on a headset. Voice recognition is very noise sensitive, and this system will be used outdoors with a lot of background street noise, so the closer the microphone is to the mouth of the user, the better signal/noise ratio. It is predicted that the action of bringing the device to the mouth is too much of an annoyance, so a headset mounted solution is chosen.

In the ViaVoice documentation and software a special ViaVoice microphone is recommended, but instead a professional unidirectional head-worn dynamic Shure SM10A microphone is used. The microphone has mechanical noise reduction, by being open on both sides towards the users mouth and away from the users mouth. This way the acoustic waves on the "away" side of the microphone is mechanically subtracted from the signal on the "user" side of the microphone, thereby reducing the background noise.

4.7 Electronic compass

Due to limitations in time and money in this project no electronic compass is included in the project. Instead the direction transmitted by the GPS receiver is used. This is far from optimal because the direction from the GPS is calculated (by the GPS receiver) by the difference between previous positions, and is thereby the direction of movement not necessarily the current heading of the user.

4.8 GPS

The GPS Receiver supplied by Bosch Telecom is used as the positioning device in this project. It is a Garmin GPS 25LP - GPS Receiver. The features that are important to this project or generally important, are as follows:

- Differential GPS (DGPS) ready.
- 12 parallel channel receiver tracks and uses up to 12 satellites to compute and update position.
- Up to one position per second. Pulse-per-second mode.
- Accuracy: without DGPS: 15m RMS (100m with SA at max), with DGPS: 5m RMS.
- Interface: RS-232 compatible with baud rates: 1200-9600.
- Output: NMEA 0183 version 2.0 ASCII output. (Comma separated lines)
- 108 different earth datums supported.

- Size: 46.5mm(w) x 69.9mm(l) x 11.4mm(h).
- Weight: 38g without cables or antenna.
- Power supply: 3.6 to 6 VDC, typical current: 115mA at 5VDC.[Garmin, 1998]

Communication with the GPS Receiver is done using the `javax.comm` packages developed by Sun Microsystems, for enabling Java programs to use serial and parallel ports for communication.

4.9 Map

In this section the production of a bitmap user map and the considerations connected to this process.

A bitmap image of a map of the center of Aalborg is produced by using an online GIS planning application named IRDSS, developed as a part of the ECC's 4th technology framework. [IRDSS-partners, 1999]

The map originates from "Kommunernes Tekniske Kortsamling", and is often referred to as T3. In urban areas it has a precision of +/− 15cm. It is produced from aerial photos, shot from a height of 750m above sea level. [Nielsen, 1999b]

The map is then thresholded for clarity. The resulting image is shown in figure 4.7.

4.10 Route planning

As described in section 3.1.1, a graph is used for planning routes from one position to a sight. In this section the production of this underlying undirected weighted graph and an algorithm for finding shortest paths between positions are described.

Nodes have no physical extent, they are considered as points. The nodes are placed in front of the sights, where a human tour guide would stand with a group and talk about the sight.

A tool for placing sight and way point nodes/vertices on the map was produced: `NodeCreator`. This tool let the developer use the mouse to click on the map and marks the spots and at the same time saves the image coordinates in the `nodes` table in the `tourist` database. In figure 4.8 the nodes and their numbers are shown.

Another tool for linking the nodes together was produced: `LinkCreator`. This tool let the user click on or near nodes to create links between them, and thereby mark the ways the ways the user can walk between the nodes. These links are saved in the `links` table in the `tourist` database. The resulting graph and underlying bitmap image is shown in figure 4.9.

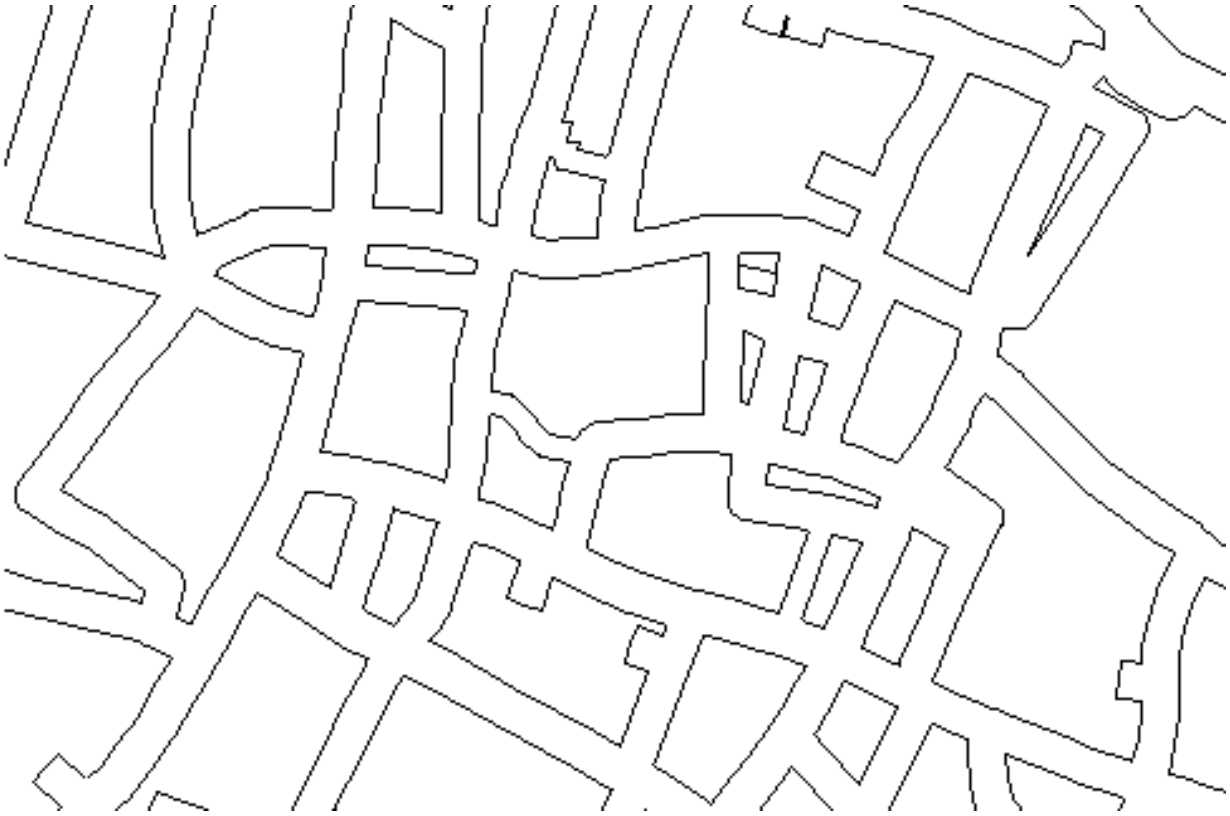

Figure 4.7: Bitmap image of map of the center of Aalborg.

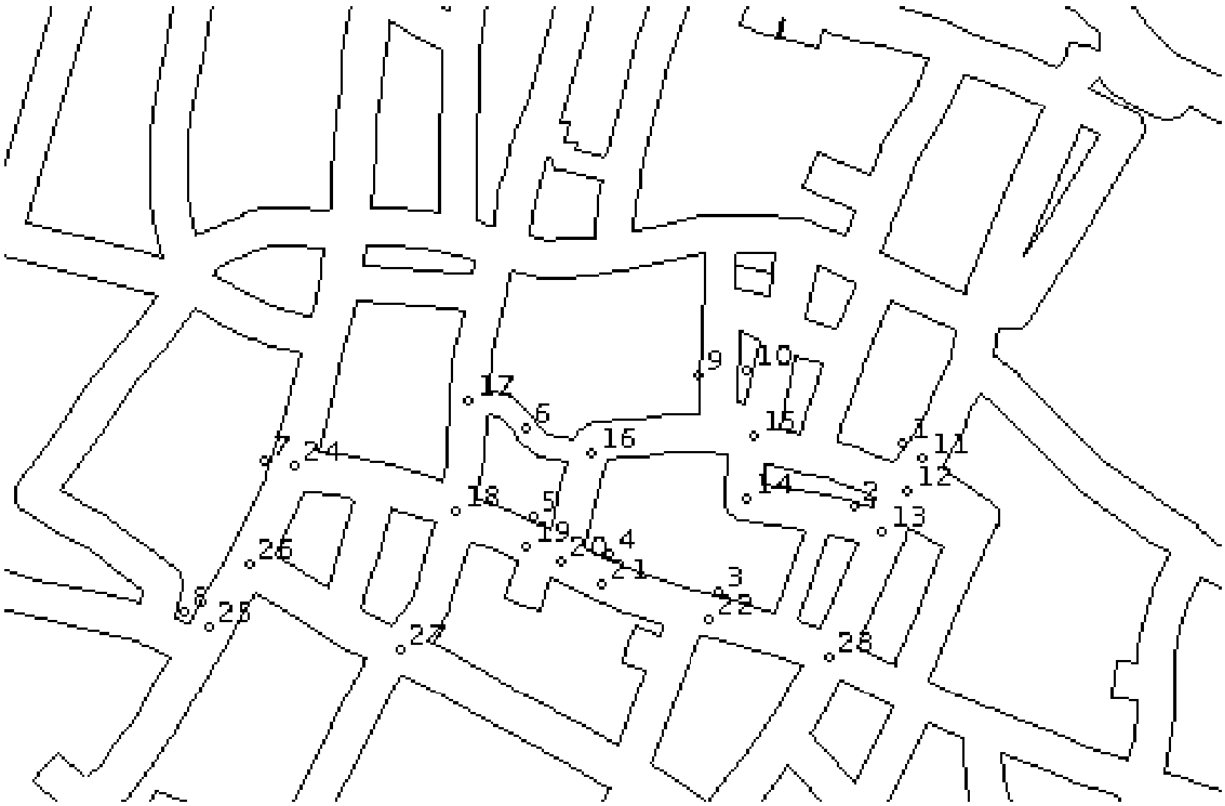

Figure 4.8: Map with nodes.

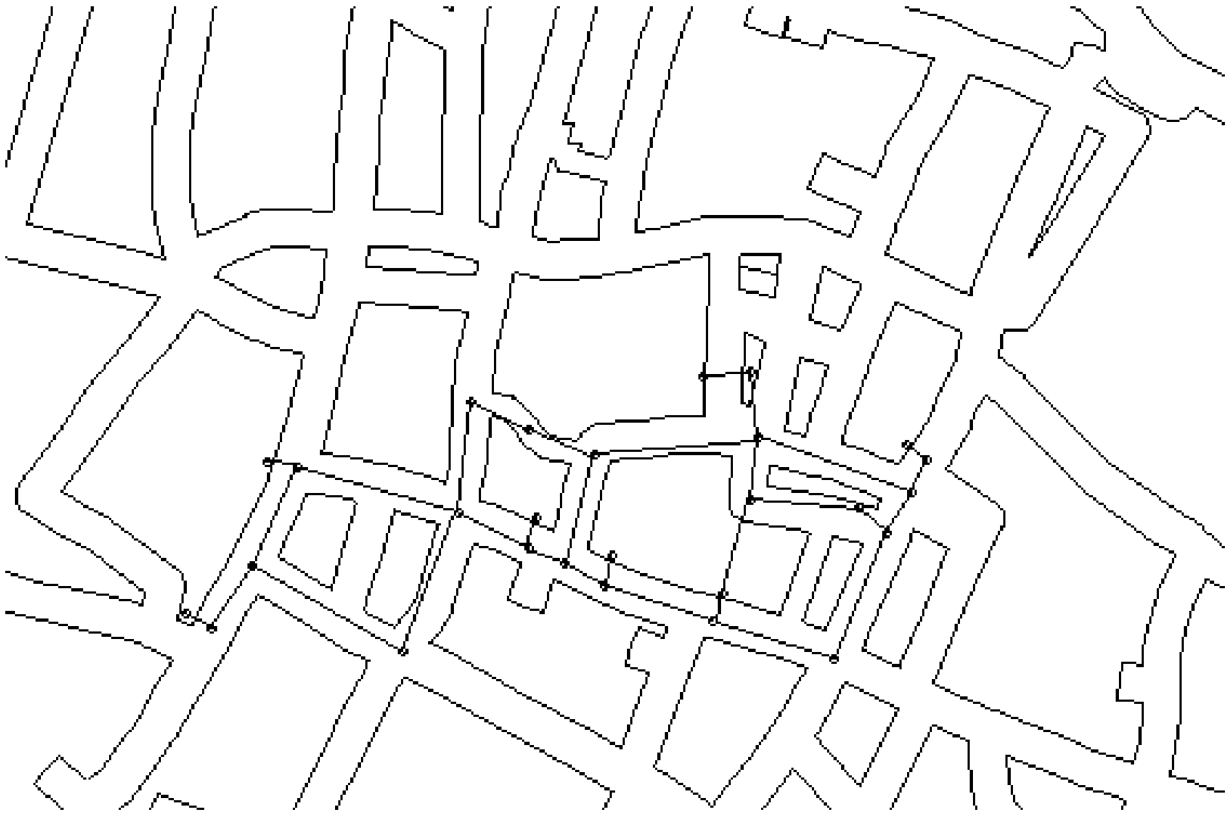

Figure 4.9: Routeplanning graph on top of bitmap image.

In the graph depicting the routes used as a demonstrator in this project: 27 nodes/vertices and 31 edges/links are used.

If all way points and sights (nodes) and routes (links) are named, these names can be used to describe the route using real-world names of e.g. roads, road junctions and squares. E.g.:

> Walk *69* meters down *Boulevarden*, to *the junction between Algade and Boulde-varden*, *turn right*, walk *66* meters up *Algade* until you get to *the Budolfi Cathe-dral.*

In this example all words in *italic* represent data selected from the database. The *69* and *66* meter distances and names of roads (*Boulevarden* and *Algade*) are selected from the `links` table. The strings: *the junction between Algade and Boulevarden* and *the Budolfi Cathedral* are selected from the `nodes` table. The *turn right* string is calculated from the angle between the two links connected to the junction. The function for constructing the direction statement, will output *turn left* if the angle between the two links is less than e.g. -45 degrees, *go straight ahead* is the angle is between -45 and +45 degrees and *turn right* if the angle is greater than 45 degrees.

A third tool for calculating the real world length of links and real-world positions of nodes was produced: `LinkWorldCalculator`. Using the transformation derived in section 4.11, the node-map coordinates are converted to longitudes and latitudes, and stored in the database. By using an algorithm from [Cederholm, 1999] for converting between geographical WGS84 coordinates and Cartesian coordinates, and calculating the Euclidean distance between nodes. These distances are in meters, and are stored in the `links` table in the `tourist` database.

Finding the shortest path involves searching for the least cost attached to each vertex in the path for the source to the destination vertex. Since this project only involves streets and paths having directions in both ways (for pedestrians anyway), the graph will not be directed. The graph can be represented as an adjacency matrix in which each element (i,j) represents a weight or length (in meters) between the two vertices v_i and v_j. A path from vertex v_i to v_j is represented by the ordered pairs: (v_i,v_k), (v_k,v_l),...,(v_t,v_j) given the restriction that no repeated vertices are allowed. The shortest route between nodes can be calculated from the adjacency matrix using Dijkstra's Shortest-path algorithm described in [Sedgewick, 1992, p.461ff]

4.11 Map- and world-coordinates

The GPS receiver produces coordinates in the world (latitude, longitude and altitude), and these coordinates have to be mapped to coordinates that can be shown on the map (map coordinates). GPS uses the WGS84 datum as does the coordinates from the reference map from Thorbjørn Nielsen.

Translation between coordinate systems can be done by using the general equation 4.1.

$$\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{gps} \\ y_{gps} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix} \tag{4.1}$$

With two positions known in both coordinate systems, scale and translation can be found. Since the image bitmap is directed with the north-south axis as the y-axis, rotation can be considered zero. Therefore only two coordinates are needed, and the equation can be simplified to equation 4.11. where image x and y coordinates refer to longitudes and latitudes, respectively. Over a small enough area, degrees can be considered as coordinates.

$$\begin{bmatrix} \frac{x_{image2}-x_{image1}}{x_{gps2}-x_{gps1}} & 0 & x_{image1} - \frac{x_{image2}-x_{image1}}{x_{gps2}-x_{gps1}} \cdot x_{gps1} \\ 0 & \frac{y_{image2}-y_{image1}}{y_{gps2}-y_{gps1}} & y_{image1} - \frac{y_{image2}-y_{image1}}{y_{gps2}-y_{gps1}} \cdot y_{gps1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{gps} \\ y_{gps} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix} \tag{4.2}$$

Four points was found in both world coordinates (longitudes and latitudes) and in image coordinates (pixels), in cooperation with Thorbjørn Nielsen. Two of these coordinate sets, was used to create the transformation matrix, and the other two sets was used to control the quality of the transformation. The two test points showed deviations of maximum 2%, so this simple method is acceptable for this project.

Chapter 5

Implementation

In this chapter the implementation effort done in this project is described. The system is implemented in part, these parts and the reason for choosing these is given.

5.1 Implemented system

The sub dialogue described in figure 4.2: Guide Single sub-dialogue, has been the aim and goal of the implementation effort. This dialogue has been chosen due to its diversity and level of complexity taking the time frame given for this project into account. The center of the implementation, and the class holding the `main` method, is the `dk.datadealeren.imm9.dialogue.GuideSingle` class. This class is build on the basis of the example(Ebberød Bank) provided by Bo Bai.

The implemented dialogue includes use of a number of devices and commands described in section 4.3.1 and 4.3.2, respectively. The following devices have been implemented, in the degree described below. All device classes belong in the `dk.datadealeren.imm9.devices` package.

GPSReceiver: The `GPSReceiver` class, can initialize and configure the GPS receiver using a serial RS-232 port on the notebook. During runtime the instance of the class holds a `getPosition` method for the central system to use. The class also holds methods and variables for telling the quality of the position (number of satellites). The system have not been developed using Differential GPS, and thereby not having the required precision. Tests have shown variations of approximately +/- 100m, which is too much to have any real value.

PDAMap: The `PDAMap` class is implemented to show the full map of the area (see figure 4.7) on the notebook screen. The communication with the PDA is not implemented, and the user interface classes for the Palm Pilot have not been developed.

PDAText: Have not been implemented.

SpeechRecognizer: Using the ViaVoice speech recognizer and the *IBM Speech for Java* implementation of *Java Speech*, the rule grammar based `SpeechRecognizer` class have been developed. A simple static grammar involving a few sights have been developed, this will be further developed to a dynamic grammar generator, importing sights from the database. The grammar can accept the user asking the system to guide him somewhere, in a number of ways.

SpeechSynthesizer: Using the *IBM Speech for Java* the `SpeechSynthesizer` class have been implemented, and is used to utter annotated sentences selected from the database.

The following commands, all placed in the `dk.datadealeren.imm9.command` package, have the following statuses:

DescribeRoute: Has not been implemented.

DescribeSight: The `DescribeSight` command uses the `Speak` command to utter the description of the sight, selected from the database.

Positioner: The `Positioner` have been developed, and communicates with the `GPSReceiver` device.

PDAList: The command used for listing sights on the PDA screen have been implemented to show a list of buttons on a Palm Pilot screen sized window(160 x 160) on the notebook screen, and accept selections made by the user.

PDAMap: The `PDAMap` command have been implemented to work with the `PDAMap` device, showing the map of the area on the notebook screen.

RoutePlanner: The `RoutePlanner` have been implemented to take two arguments: Source and Destination nodes. After calculating the shortest route using the graph constructed from the database, a sequence of nodes are returned encapsulated in a `Vector` object.

Speak: The `Speak` command, used to utter sentences using the `SpeechSynthesizer` device, is implemented without the option of interrupting the utterance.

SpeechListen: The `SpeechListen` command is implemented taking a Speech Recognizer, a JSGF-grammar and a timeout value as inputs, returning the tag from the grammar rule that fired, or an empty tag if no rule fired before the timeout.

SpeechAndPDAListListen: Has not been implemented.

Switch: The `Switch` command used was developed for the example dialogue by Bo Bai.

Tester: The `Tester` command holds a number of debugging utilities.

The domain knowledge module have been implemented using a database and a number of tables:

categories: Holding a number of category ID's and names, used to group sights.

links: Holds the links between the nodes in the graph: Link ID's, two node ID's, the distance between them and the name of the link.

Nodes: Holds the way point- and sight nodes, ID's, map- and world coordinates of each node.

SightCatRel: Relations between sights and categories.

Sightnodes: Relations between sights and nodes.

Sights: Hold the sights included in the system, with: ID's, names, descriptions, keywords, building year. As more attributes to the sight are added, extra fields will be added.

The dynamic knowledge module managing the semantic frames has not been implemented.

Chapter 6

User tests

Since the implemented system is incomplete, no user tests have been conducted. Instead an outline of testing the full designed system is described in this chapter.

A Validation test could be conducted presenting potential users with a number of tasks to be completed using the system. Tasks would take the form:

> "Using the system as your only mean for directions, get to Budolfi Cathedral from where you are now." - presenting users with this from the same spot in each test.

The user would be asked to think aloud, saying to the test monitor walking along with him, what they expect from the system when they are using it. After performing the tasks the user would be presented with a questionnaire with sections of Likert statements concerning the performed tasks and a section of statements concerning the system as a whole. These questionnaires can be used to quantify the results, and perform statistical analysis.

This type of system poses a problem in documenting the tests, because they do not take place at one location. The system could keep a log, recording what the user says. But the users physical actions, like pointing to a sight, can not be documented unless a hand held video camera is used.

The test users can be found by contacting Aalborg Tourist Council, and asking them to direct some tourists that wants to go on a tour of the center of Aalborg to the test team. Kaj Handberg CEO of Aalborg Tourist Council has expressed great interest in the development of new tourist aids and tools. [Rubin, 1994]

Chapter 7

Conclusion

The purpose of this project has been to learn about Advanced Intelligent MultiMedia systems, this has been achieved by analyzing existing information material aimed at tourists in Aalborg. On the basis of this analysis and some of the ideas of the developer and others, a position aware speech enabled hand held tourist information system have been designed. This designed system have been partly implemented, and an outline for a user test of the full designed system have been presented. The designed and partly implemented system is a very complex system with a lot of different very advanced aspects and complex problems to solve.

Position awareness needs DGPS to be accurate/precise enough in systems used by pedestrians. Position awareness in a speech enabled PDA systems introduces a huge number of possibilities of presenting the user with more precise information due to geographical relevance.

In the future when the CPK Dialogue Manager is fully developed, it will enable easy development of dialogue systems. The command structure enables reuse of large amounts of complex code, even commercial commands without source code available, can be used because of the Java Reflection technology.

Chapter 8

Future development

In this chapter future development opportunities to the system will be discussed. The system could be developed further at a number of accounts, here a solution with an Advanced Route Planning module is introduced.

8.1 Advanced Route planning

An individual route planning subsystem could increase the flexibility of the system, and the tourists would not have to follow predefined tours. The route planning user interface could look as the draft in figure 8.1.

In the lower of the screen shots in figure 8.1 the user has an opportunity to select more sights or to delete sights that are too much out of the way. The system can while the map is shown inform the user of the length of the tour in meters and an estimate of the duration of the tour.

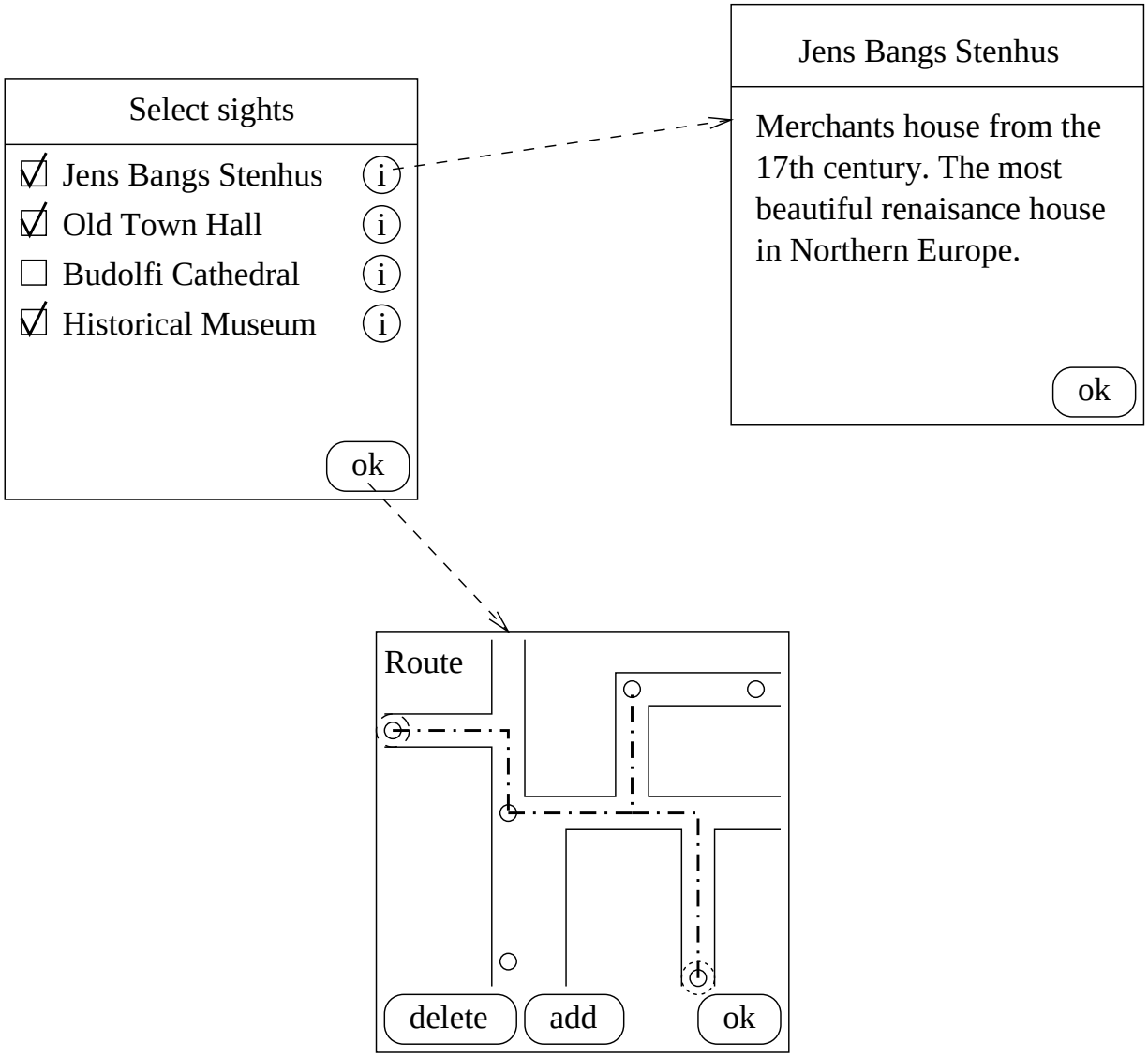

Figure 8.1: Future individual route planning screen shots

Appendix A

Scenario based usage analysis

In this appendix a scenario based usage analysis is devised. The two scenarios below, contains a number of situations the system can be used in, and all the three modes described in chapter 1 are used.

A.1 Scenarios for dialogue design

A.1.1 Scenario 1: Show me the way to ...

User<speak>: Show me the way to the church!

System<speak>: Which church do you mean?

System<screen>: Lists the nearest 5-8 entries of churches

User<speak>: Budolfi

`or`

User<screen>: Selecting Budolfi by tapping on the item in the list on the PDA screen.

System<speak>: There are approximately 150m to Budolfi church. To go there you have to turn right, and go 100m down Boulevarden, then turn left at the lights and walk 50m down Algade. Then the Budolfi Church is at your right.

System<screen>: Shows the destination in the top of the screen and shows the route on a map as a dotted line together with an arrow indicating the direction to the final goal. The size of the arrow can be used as a hint of the distance to the goal. When the user starts walking his progress on the route is plotted with a full line. If the user walks off the planned route, a new one is calculated, and shown.

...later...

User<speak>: Please tell me the way to the church, from where I am now.

System<speak>: Go 30m straight ahead, turn right at the lights and walk down Algade 50m. Then the Budolfi Church is on your right.

System<screen>: Shows the new route on the map.

A.1.2 Scenario 2: Guided tour and self exploration

User<speak>: Please guide me on a tour around this part of town.

System<speak>: How long a tour do you want?

User<speak>: Shorter than 10km.

System<speak>: OK, If you start the tour by turning right and start walking down Bispensgade, I will tell you what there is to see. To your right the old bank building ...

System<screen>: Shows facts about the current object/place being presented. Ex. Name, Address, Building year...

...later...

User<speak>: What is that?

User<pointing with the device>: to Jens Bangs Stenhus

System<speak>: That is Jens Bangs Stenhus. It was built in the 17th century by a rich merchant named Jens Bang.....

System<screen>: Shows facts about Jens Bangs Stenhus.

...a little later...

User<speak>: Thanks, that is enough.

System<speak>: Would you like to continue the tour from here?

User<speak>: Yes, please!

System<speak>: Just to the south of Jens Bangs Stenhus the old city hall is located. If you walk straight ahead for 20m, and turn right, you can see the beautiful facade of the building....

System<screen>: Shows facts about the old city hall

...later...

User<speak>: Please end the tour, and take me back to where I started.

System<speak>: Sure, I hope you enjoyed the tour. There is approximately 1.5km to the starting point. You have to turn left and walk.....

System<screen>: Shows the route back to the starting point.

A.2 Dialogue components

In the sample dialogues above two types of user utterances/actions are identified:

Commands: eg. "Show me the way to ..."

Answers: eg. "Shorter than 10km", Choice in list on PDA screen.

Three different types of system utterances/actions are identified:

Questions: eg. "Which church do you mean?" and presenting list on PDA screen.

Guiding: eg. "There are approximately 150m to ..." and shows map with directions on PDA screen.

Describing sights: eg. "That is Jens Bangs Stenhus, It was built..." and showing facts on PDA screen.

To describe the sentences recognized by the system, the Java Speech Grammar Format(JSGF) is used. This is chosen because the Java Speech API is planned to be used in the system. [Sun-Microsystems, 1998]

A.3 Commands

Vocal commands used by the user in the example dialogue:

1. *Show me the way to the church.*
2. *Please tell me the way to the church, from where I am now.*
3. *Please guide me on a tour around this part of town.*
4. *What is that?*
5. *Thanks, that is enough.*
6. *Please end the tour, and take me back to where I started.*

A.3.1 Command type 1 and 2: ...the way to...

Command #1 and #2 have the same effect:

- The system starts a new mode, storing the position in the new parent mode.
- The system checks if it has all necessary information: Current position and end position.
- If not, it asks for the missing information.
- When the system has all the information needed, it calculates a route to the goal, show the route on the PDA screen, and starts guiding the user to the goal.

This type of commands can be generalized to the form:

```
<polite>* <action> the way to <sight> [position] <polite>*
```

Where `<polite>*` is an optional *please*, *could you*, *computer* or combinations of these, eg. *Computer, could you please ....*

`<action>` is: *show*, *show me*, *tell me*, *what is...*

`<sight>` is one of all the names of sights the system knows, or one of their types. eg. *Budolfi Church*, *Budolfi Cathedral*, *the Cathedral.*

`[position]` is the optional: *from where I am now* or *from my current position.* This information is really irrelevant to the system, but it makes the sentence uttered by the user sound more natural in some contexts. In a more general system a function for calculating routes between goals/sight from other positions than the current, would be included.

A.3.2 Command type 3: ...a tour...

In a similar way command #3 can be described by:

- The system starts a new mode, but not storing the position in the previous mode. The old mode is not stored due to the fact that a guided tour is of another nature that directions to a single sight.
- The system checks if it has all necessary information: Current position and maximum duration or length of tour. This could also include asking for some of the users preferences of sights, but this is not included here.
- If not, it asks for the missing information.
- When the system has all needed information it finds the nearest sight that is included on one of its tours and the starts the tour from there. This implies that all tours should be of a cyclic nature.

Tour starting commands can be generalized to:

```
<polite>* <action> a tour [position] <length> <polite>*
```

Where `<action>` is: *take me on*, *give me*, *start...*

`[position]` is the area where the tour is wanted. eg. *around Nytorv*, *south of the harbor*

`<length>` can be: *for* `<X>` *minutes*, *shorter than* `<X>` *kilometers.*

A.3.3 Command type 4: What is that?

By combining an electronic compass, a GPS receiver and a GIS type database structure, the system can calculate what the user is pointing the device at. This functionality is activated by this command. The effect of the command is as follows:

- Are current position and direction known?
- If not, get these values from external devices.
- By extending a line from the current position in the current direction, the nearest sight in this approximate direction can be found.
- This implies that a sight always will be found, but it can be very far away from the direction the user pointed.
- If the system finds two sights that are equally likely to be the one the user is referring to, it have to ask which one the user is interested in.
- When the sight is determined, the system reads the description associated with the sight, and shows the description on the PDA screen.

In a more general system the system should be able to identify streets, squares and other landmarks.

```
What is that?
```

A.3.4 Command type 5: interruption

If the system is reading a longer text, the user should always have the possibility to stop the system. The system should only stop reading the current text, nothing else.

```
<polite>* [prewords] <meaning> <polite>*
```

Where the optional `[prewords]` can be: *That is*, *Now you can*,...

And `<meaning>` can be: *stop*, *enough* or *halt.*

A.3.5 Command type 6: ...end...

When the system is guiding the user to a destination or the user is on a guided tour, the user always have the option to abort the action, by giving this command.

```
<polite>* [prewords] <meaning> the tour <polite>*
```

Where the optional `[prewords]` can be: *Now you can* or *Now I would like to.*

`<meaning>`: can be: *end* or *finish.*

A.4 Answers

The answers the user gives to the system in the sample dialogue can be listed as:

1. *Budolfi*
2. *Shorter than 10km*
3. *Yes, please!*

A.4.1 Answer type 1: picking a sight

If the system is in doubt what sight the user i referring to it should put together a list of maximum five to eight choices of the most likely sights. These should alway be ordered by the knowledge the system have, and thereafter the distance to the sights. Eg. the user asks the system to show him the way to the church. Then the system finds the five nearest churches, and shows them on the screen. The user can select an item in the list on the PDA or speak his selection. The expected input here is both very simple and very diverse:

```
<sight>
```

A.4.2 Answer type 2: Defining distance or duration of a tour

This can be very tricky because the input from people talking about distance or timely duration, are very soft and can be said in many different ways.

```
[relative] <X> <length>
```

Where the `[relative]` term can be: *Shorter than* or *Not longer than...*

`<X>` is a number, including time related sizes, like *quarter of a hour.*

The `<length>` term defines whether the measure of the tour is time or distance. It can be: *m*, *km*, *minutes*, *hours*

A.4.3 Answer type 3: Confirmation or rejection

It is sometimes needed to get explicit confirmation from the user, like the following:

```
<polite>* <bool> <polite>*
```

Where the `<bool>` can be: *yes*,*no* or equivalents.

A.5 Questions

The questions asked by the system to the user in the sample dialogue can be listed as:

1. *Which church do you want?* + Presenting a list of choices on the PDA screen.
2. *How long a tour do you want?*
3. *Would you like to continue the tour from here?*

A.5.1 Question type 1: resolving ambiguities

If the system is in doubt of which sight the user is referring to, it should ask a question to resolve the ambiguity. This is done by finding the closest sights matching the information the system already has. This is practically done by doing a search in the domain knowledge database, using the words and categories the user has given to the system already. The closest matching 5-8 sights is then listed on the PDA screen, at the same time the system asks the user which sight he is referring to.

A.5.2 Question type 2: requesting missing details

If the system lacks one or more pieces of information to be able to return the requested information, it will ask a question to get the missing information. This is done using a custom fitted semantic frame metaphor. [Brøndsted et al., 1998]

A.5.3 Question type 3: requesting course of action

When the system has finished a task and thereby ended the submode, it will ask the user if he wants to continue the parent mode, that was left when the user requested the system to go into the submode. The current mode can be visualized to the user by an icon on the PDA screen.

A.6 Guiding

In the sample dialogue, the system are guiding the user like this:

1. *There are approximately 150m to Budolfi church. To go there you have to turn right, and go 100m down Boulevarden, then turn left at the lights and walk 50m down Algade. Then the Budolfi Church is at your right.* + Show map with directions on PDA screen.

2. *Go 30m straight ahead, turn right at the lights and walk down Algade 50m. Then the Budolfi Church is on your right.* + Show map with directions on PDA.

3. *OK, If you start the tour by turning right and start walking down Bispensgade, I will tell you what there is to see.* + Show map with directions on PDA screen.

4. *Sure, I hope you enjoyed the tour. There is approximately 1.5km to the starting point. You have to turn left and walk.....* + Show map with directions back to starting point on PDA screen.

A.6.1 Guiding type 1+2: Directions to one specific place

This type of system response is used when the user have given the system a command to direct him to a specific sight, or the system is guiding the user between sights on a tour of the town. This is done by finding the route described in chapter 4.10. The uttered instructions are constructed by sentences telling the user to go a number of meters ahead, and telling the user to turn in a certain direction.

In parallel to the spoken directions a map with the route to the next sight is shown on the PDA screen. This is continuously updated as the user moves.

A.6.2 Guiding type 3: Starting a guided tour

This type is used when the user has requested a tour of the town. At this time the system has all the information that it needs. The current position is saved so the system can guide the user back to this point. The nearest sight on the selected tour is found, and the user is guided there. When the user arrives to the sight the system describes the sight, and guides the user to the next sight, and so on. Then on a tour, the system all the time shows the name of the current sight and facts about this, and the name of the next sight.

A.6.3 Guiding type 4: Ending a guided tour

When the user have completed or prematurely ended a tour, the system speaks a sentence like *"I hope you have enjoyed the tour. To get back to our starting point, you have to ..."*,

giving the user the directions to get back to the starting point, both spoken and on the PDA screen.

A.7 Describing sights

In the sample dialogue, the system is describing to the user what he sees with these sentences:

1. *To your right the old bank building ...* + Shows facts about the current object/place being presented. Ex. Name, Address, Building year on the PDA screen.
2. *That is Jens Bangs Stenhus. It was build in the 17th century by a rich merchant named Jens Bang...* + Facts of Jens Bangs Stenhus on PDA screen.

A.7.1 Describing sights type 1: Simple description of a sight

The descriptions of sights are found in the knowledge database, and are spoken to the user. Likewise facts about the current sight is shown on the PDA screen.

A.7.2 Describing sights type 2: Describing sight on request

This type is very similar to the previously described. The only difference is that the beginning of the description is formed on a form that replies to the users question. This first sentence is on the form:

```
That is <sight>.
```

A.8 Problems - out of category

1. *If you walk straight ahead for 20m, and turn right, you can see the beautiful facade of the building....* + Shows facts about the old city hall on PDA screen.

This sentence is both a guiding and a description of a sight. This type of sentences do not fit into my model of utterances, so it will not be considered any further. But this situation shows an important limitation of the dialogue structure of the system, that can be formulated quite clearly:

> Each sentence or action used in the dialogue between the user and the system is limited to apply to one and only one action.

In a future version of the system the Dialogue Manager should concatenate a number of replies into a single utterance.

A.9 Generalized user utterances

The user utterances recognized by the system can be generalized to the following. Although this is a generalization of all sentences recognized by the system, far from all sentences that can be recognized in the template make sense. This template is intended for use with the speech recognizer, not a natural language parser. The template is produced to make a single template, compared to having several different templates.

```
<polite>* <bool>* <prewords>* <action>* <meaning>* <sight>* <position>*
                    <relative>* <length>* <polite>*
```

`<polite>`: *please, could you, computer, guide, NAME OF SYSTEM*

`<bool>`: *yes, no, yep, nope, ok, I will, correct*

`<prewords>`: *that is, now you can*

`<action>`: *take me on, give me, start*

`<meaning>`: *the way to, a tour, end, finish*

`<sight>`: Any of the sights known by the system

`<position>`: *around* `<sight>`, `<direction>` *of the* `<sight>`, *from where I am now, from my current position, here*

`<direction>`: *north, south, east, west, back, forward*

`<relative>`: *shorter than, not longer that, around*

`<length>`: any number *m, km, minutes, hours*

With this template all sentences uttered in the sample dialogue, can be recognized.

Bibliography

[AB, 1999] AB, T. D. (1999). Mysql. http://www.mysql.com.

[Bækgaard, 1996] Bækgaard, A. (1996). Spoken language dialogue systems - a generic dialogue system. Technical report, Center for PersonKommunikation, Aalborg University.

[Borre, 1995] Borre, K. (1995). *GPS i landmålingen*. Ålborg.

[Brøndsted et al., 1998] Brøndsted, T., Dalsgaard, P., Larsen, L. B., Manthey, M., McKevitt, P., Moeslund, T. B., and Olesen, K. G. (1998). A platform for developing intelligent multimedia applications. Technical report, Center for PersonKommunikation, Aalborg University.

[Cederholm, 1999] Cederholm, P. (1999). Professor, institute 4, aalborg university. http://www.i4.auc.dk/pce.

[Computer, 1999] Computer, Z. (1999). Advertisement.

[Council, 1999a] Council, A. T. (1999a). Aalborg guide, 1999, danish.

[Council, 1999b] Council, A. T. (1999b). Velkommen til aalborg. http://www.aalborg-turist.dk.

[Dalsgaard and McKevitt, 1998] Dalsgaard, P. and McKevitt, P. (1998). A frame semantics for an intellimedia tourguide. Technical report, Center for PersonKommunikation, Aalborg University.

[Danielsen, 1999] Danielsen, P. (1999). Internet på farten. *Alt om net*, nr. 6.

[E-Studyboard, 1999] E-Studyboard (1999). Master of intelligent multimedia. http://www.esn.auc.dk/masters/multimedia.

[Garmin, 1998] Garmin (1998). Garmin gps 25 lp series - technical specification. Technical report, Garmin.

[Gessler and Kotulla, 1995] Gessler, S. and Kotulla, A. (1995). Pda's as mobile www browsers. *Computer Networks and ISDN systems*.

[Hesselgrave, 1999] Hesselgrave, J. (1999). Infovision project manager, am production multimedia k/s, aalborgtvr. Interview.

[IBM, 1999] IBM (March 1999). Smapi developer's guide - ibm viavoice sdk. Technical Report version 1.0, IBM.

[IRDSS-partners, 1999] IRDSS-partners (1999). Irdss. http://www.vbn.dk/irdss/aalborg.html.

[Kjærgaard, 1999] Kjærgaard, D. (1999). Med nettet i baglommen. *Datatid*, Oktober:66.

[KMS, 1999] KMS, K. o. M. (1999). Gps og dgps. http://www.kms.min.dk/geodesy/gps.htm.

[Ølholm, 1999a] Ølholm, M. (1999a). Ppc'er fra compaq og hp. *Alt om Data - Eksklusiv*, November:24.

[Ølholm, 1999b] Ølholm, M. (1999b). Småt, men godt fra psion og handspring. *Alt om Data - Eksklusiv*, Oktober:26.

[Mathiassen et al., 1993] Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., and Stage, J. (1993). *Objektorienteret analyse*. Forlaget Marko Aps.

[Nielsen, 1999a] Nielsen, J. K. (1999a). *Road Pricing - en prototype*. Afgangsprojekt fra Landinspektør uddannelsen, Aalborg Universitet.

[Nielsen, 1999b] Nielsen, T. (1999b). Gis librarian, institute 4, aalborg university. Interview.

[Nordjysk-Informatik-Forum, 1998] Nordjysk-Informatik-Forum (1998). Gis på nettet.

[Olesen, 1998] Olesen, S. B. (1998). Good old aalborg.

[Pedersen, 1999] Pedersen, K. (1999). Palm-computer den mest udbredte håndholdte pc. http://www.comon.dk/nycomon/udskrift.idc?id=2962.

[Rubin, 1994] Rubin, J. (1994). *Handbook of Usability Testing*. Wiley Technical Communication Library.

[Sedgewick, 1992] Sedgewick, R. (1992). *Algorithms in C++*. Addison-Wesley.

[Sun-Microsystems, 1998] Sun-Microsystems (1998). Java speech grammar format specification. Technical report.

[Sun-Microsystems, 1999a] Sun-Microsystems (1999a). Java speech api. http://www.javasoft.com/products/media/speech.

[Sun-Microsystems, 1999b] Sun-Microsystems (1999b). Kvm - java for palmos. http://www.javasoft.com/products.

[Wabasoft, 1999] Wabasoft (1999). Waba - "java" for palmos and wince platforms. http://www.wabasoft.com.

[Walsh and Daly, 1998] Walsh, D. and Daly, P. (1998). Precise positioning using glonass. *GIM - Geomatics Info Magazine*, November.

[Wellhausen, 1989] Wellhausen, H. (1989). Electronic compass with direction-sensitive magnetic field sensors. *Elektronik*, Vol. 38 Issue 8:85–8.