The CityInfo Pedestrian Information System

An Experiment with Content Creation and Presentation Techniques

Volker Paelke, Birgit Elias and Mark Hampe
Institute of Cartography and Geoinformatics
University of Hannover
Appelstr. 9a,30167 Hannover
Germany
Volker.Paelke@ikg.uni-hannover.de
Birgit.Elias@ikg.uni-hannover.de
Mark.Hampe@ikg.uni-hannover.de

Abstract

As the technical prerequisites for location based services (LBS) become widely available the development of practical services and the creation of the content required for them becomes a pressing question. Initial experiences show that content creation for location based services differs significantly from established multimedia creation for PC and web-based applications due to differences in the user interface and constraints imposed by hardware performance, hardware form factor and the available software and operating system architectures. This paper reports on a project in which different content creation and presentation techniques are evaluated in a practical setting to derive guidelines for developers.

Motivation

Advances in mobile computing and wireless communication technology now enable the creation of interactive multimedia applications on a variety of mobile devices ranging from mobile phones over smart phones to PDAs (Personal Digital Assistants) and other portable computing devices. Applications like location based services, mobile e-Commerce applications (m-Commerce) and multimedia entertainment are envisioned as future mass-market applications that can be created based on these technologies. However, the design of multimedia content and user interfaces for mobile devices is currently complicated by a lack of standardized visualization techniques suitable for small displays, the absence of adequate authoring tools to enable content creation by non-technicians and a lack of guidelines and style guides for content creators. A number of recent conferences and workshops have examined these areas (e.g. Dunlop and Brewster, 2001; Johnson, 1998; WMCSA, 2002). Despite advances in several areas the creation of attractive and usable LBS content is still hindered by a lack of related design experience, guidelines, processes and corresponding tools. Key problems in the design of interfaces for mobile applications are the development of appropriate visual presentation designs because these differ significantly from the well known desktop and web domains. Like cartographic presentation techniques the presentation techniques employed in a LBS system should fulfill several criteria, e.g. expressiveness (all information should be contained in the presentation) and effectiveness (the information should be sufficiently clear and understandable). For practical applications the overall appropriateness of a presentation technique covers not only expressiveness and effectiveness but also the cost incurred by the creation of the corresponding content (e.g. a slightly less effective technique could be preferable if content creation is significantly cheaper or faster). The methods used for content creation can vary widely from programming over authoring and semi-automatic acquisition to automatic generation from databases. They also differ significantly in the time, expertise and cost required for content creation. However, one key difference to classical cartography is that there is currently very little information to guide developers in choosing appropriate techniques and tools.

Related Work

In the last few years a lot of prototypes for LBS applications on PDAs, mostly the HP iPAQ series, are developed to investigate various research aspects: different research communities focused on topics like system architectures, telecommunication, human-computer interaction, cartographic presentation techniques and these issues are often mixed up in one project. Most of them have a special use case in mind (in general touristic settings for pedestrians like city guides and hiking) and deal with client-server architectures. Here only a selection of projects is referenced.
The research project GiMoDig deals with transmitting geodata from different National Mapping Agencies in real-time on PDA providing user adapted map layouts. Additionally, issues like system architectures and real-time generalization using MRDB are dealt with (Sarjakoski et al., 2002) (Elias, Hampe and Sester, 2005). Some projects stress the issues of limited technical resources of mobile navigation systems and develop resource-adaptive pedestrian navigation systems for outdoor and indoor (REAL) (Baus, Krüger and Wahlster, 2002) or positioning methods for pedestrians in city environments (NAVIO) (Gartner, Frank and Retscher, 2004). Others concentrate on aspects like design and usability

ISBN: 0-958-46093-0

testing, e.g. in the SAiMotion project (Heidmann Hermann and Peissner, 2003). Prototypes integrating 3D information are introduced for example with DeepMap, a tourist guide for the city of Heidelberg (Malaka and Zipf, 2000), and TelMaris combining 3D-models with 2D maps to provide boat tourist with information about cities round the Baltic sea (Schilling, Coors and Laakso, 2005). Similar to the project presented here, the LoL@ City Guide supplies a Vienna tourist guide exploring especially the cartographic portrayal techniques on small displays (Gartner and Uhrlirz, 2001). Furthermore, it is intended to study the technical feasibility of mobile applications using UMTS and provide the users the chance to exploit the capabilities of these new mobile communication systems (Pospischil, Umlauft and Michlmayr, 2002). An overview about the existing map-based mobile guides is presented in (Baus, Cheverst and Kray, 2005).
A quite different approach to provide pedestrians with navigation information presents (Kolbe, 2004), using geocoded videos and panoramas augmented with additional information like street names, orientations and guiding instructions.

Approach

The CityInfo project aims to gather practical information on both the cost of creation and the effectiveness of multimedia presentation techniques for a typical range of LBS content. To achieve this, the CityInfo system combines results from software engineering, human-computer interaction and cartographic presentation techniques into a testbed application for Pocket PC PDAs. Different approaches to content creation and presentation are used in an iterative development process to create content for an interactive tourist guide that is evaluated with real users.
The dissemination of multimedia content from content creators to end-users by means of a LBS system consists of three activities: Content creation, content management and delivery, and content presentation (see Figure 1).

Figure 1: Dissemination of Multimedia Content

Each of these activities must be supported by adequate software.
Content creation is typically achieved by using a set of software tools for designing or capturing multimedia data, editing multimedia raw-data and encoding it in a suitable data format. Two approaches to the creation of multimedia presentations and content management are in widespread use today: The first consists of high-level toolkits that present the required multimedia management and presentation functionality at a higher level of abstraction to programmers while the second category takes the form of (mostly visual) authoring tools. For web- and desktop-based multimedia applications authoring tools that enable content creators to combine the individual media-elements into a coherent multimedia presentation without the need for in-depth programming are used primarily. Most authoring tools also provide a high-level scripting language to enable the creation interactive multimedia content. Experiences in multimedia development projects for desktop and web platforms have shown that while high-level toolkits succeed in making the programming of multimedia applications easier and faster for technology experts, they are still insufficient to bridge the technology gap for average multimedia content developers.
For LBS applications on mobile devices the situation is somewhat different: While existing software tools for the creation of multimedia content can in principle be applied there is a need to address the specific requirements of mobile devices, especially the limitations of available screen space (that necessitates specific graphical presentation techniques), limited memory and processor resources (that restrict the size and complexity of multimedia content) and the data formats supported by mobile devices. It is one aim of the CityInfo project to evaluate existing tools with regards to these specific requirements. For authoring tools the situation is even more exposed. While a few authoring tools have been developed for mobile applications in research projects there are few examples in which these have actually been applied by real content developers outside the project context in which they were developed. Closer examination of existing authoring tools in a pilot study showed that they are typically based on the well-intended intuition of computer scientists and fail to address the real requirements of multimedia content developers (Geiger, Paelke, Reimann, and Stoecklein,, 2002). The two main issues raised by potential content developers were that the tools were either elaborate visual interfaces to the base software that failed to support a content centred way of work and only provided a different presentation of the underlying software libraries or they were indeed completely visual editors that in theory were suitable to support a content centred workflow, but were in practice limited to the creation of simplistic toy applications in a very limited application domain. Since no general purpose authoring tools for LBS currently exists it is one of the aims of the CityInfo project to establish specific requirements for such authoring tools.
The content presentation activity also differs between desktop and web-based platforms on one side and mobile platforms on the other. While elaborate run-time systems for multimedia content presentations exist on desktop platforms (often as browser plug-ins or as run-time environments of the corresponding authoring tools) the support for

content presentation on current mobile devices is quite limited. Typically, limited presentation capabilities for 2D graphics, sound and text are provided through operating system functions while more complex media formats like video and interactive 3D graphics require additional software packages. The use of all of these presentation function requires some form of programming.

To establish the requirements that content creators place on authoring tools necessitates the same user-centred approach to their creation as to other interactive systems. This means that requirements must be identified by studying the work processes and tasks of the intended users – in this case the different stakeholders in development process and their respective tasks. Although this approach may seem obvious, surprisingly few design methods and tools published in literature refer to a previous user study. A notable exception in the domain of graphical user interfaces is the study of Rosson, Kellog and Maas (1988) in which structured interviews were used to gather information about design practice and perceived problems during the design process. For the creation of mobile multimedia authoring systems this gives rise to a hen-and-egg problem: On one side the lack of appropriate authoring tools precludes most potential content designers from creating application content. On the other side few potential content designers are familiar enough with the potential and constraints of mobile multimedia technology to explicitly state their requirements for an authoring tool. To resolve this problem we use an approach consisting of three activities:

The first activity is to implement the CityInfo system as an exemplary LBS application with a wide range of multimedia content using the basic capabilities provided by the Pocket PC operating systems and existing tools for content creation. The second activity gathers the experiences of the content creators through observation and interviews to serve as guidelines for future content development activities and to derive the actual requirements that content creators place on an authoring tool for LBS content. The third activity will be the creation of appropriate authoring tools based on the requirements identified but is not covered in this paper.

The observation derived from this development project can be applied by content creators to choose appropriate techniques and tools from those currently available and will eventually be used to define the requirements for future authoring tools for LBS.

Testbed: The CityInfo System

The use scenario in CityInfo is an interactive version of the “red-thread”, a tourist walk through the downtown area of Hanover that passes 36 points of interest (POIs). It was chosen to create an application that appeals to a large population of casual users in order to enable meaningful tests with different presentation modalities (e.g text, audio, 2D raster and vector graphics, 2D video, 2D vector animation and interactive 3D animation) within a realistic setting with a wide range of potential users.

There are two major hardware platforms available on which truly mobile LBS applications can be implemented: Personal digital assistants (PDAs) and Smartphones. PDAs are typically larger than Smartphones and have traditionally offered more computing performance, but the market of Smartphones is already larger than that for PDAs and the dynamic development of Smartphone hardware suggests that they will match PDA capabilities in the near future and might eventually replace them completely. To anticipate future hardware developments we have selected one of the most powerful PDAs currently available as the reference platform for CityInfo: the HP iPAQ hx 4700. The iPAQ hx 4700 features a high resolution 4-inch display with 480x640 pixels and an Intel XScale Processor with 624 MHz and operates under the Windows Mobile 2003 SE operating system (also referred to as Pocket PC 2003). To provide localization information a Holux GR-230 GPS receiver is used that communicates with the PDA through a wireless Bluetooth connection. The wireless connection operates with a range of up to 10 meters and allows to position the GPS receiver separately from the PDA, which helps to enhance GPS reception and allows for comfortable handling of the handheld PDA without a bulky GPS module.

As discussed in the previous section the lack of established multimedia run-time systems for mobile devices makes it necessary to program the content management and content presentation functionality of a LBS from scratch. The choice of the implementation platform is the first major design decision for LBS developers. The main contenders for Pocket PC PDA devices are the micro edition of the Java 2 Platform (J2ME) and Microsoft’s .NET Compact framework. As part of a pilot study both approaches were evaluated with regards to their suitability for CityInfo. Both platforms are based on proven technologies, support cross-platform development and allow to leverage existing developer know-how from desktop applications through known tools and APIs. The .NET compact framework is a lightweight version of Microsoft’s .NET framework that supports a limited selection of the original functionality and is accessible from the vendor specific programming languages Visual Basic (VB) .NET and C#. J2ME is an edition of the Java 2 platform that is specifically tailored to mobile devices. It can be further categorized into configurations and profiles. For the implementation of a LBS like CityInfo the CDC – Connected Device Configuration – is the most relevant profile. It covers Pocket PC PDAs as well as Smartphones (e.g. device with the Symbian Series 80 operating system and UI) and is also available for PCs and most other mobile devices under various operating systems, thus making true cross-platform development possible. The following table (Figure 2) provides a brief overview of the features and differences of the two platforms.

	Java 2 ME	.NET compact framework
Supported operating systems	all major mobile platforms	Microsoft Pocket PC / Win CE
Supported programming languages	Java	Visual Basic .NET, C#
Availability /Licenses	various JVMs by different vendors	pre-installed on current Pocket PC PDAs
Development Tools	various Java IDEs and command line tools	Microsoft Visual Studio
Specification	Open community process, large amount of literature available	Microsoft, integrated into .NET documentation
Available APIs	Subset of J2SE APIs, mobile specific extensions	Subset of .NET
UI Elements	AWT, various widget sets	limited subset of .NET controls
Multimedia Support	Subset of JME	access to WIN32 operating system functions via P/Invoke, access to MediaPlayer
LBS Support	limited on-device support, support for web-services	no on-device support, support for Microsoft MapPoint webservice

Figure 2: Overview of Features and Differences of Java 2 ME and .NET compact framework

Both approaches have their benefits and shortcomings. While the Java-based approach allows easy cross-platform development and skill-transfer from desktop and web development the lack of a pre-installed Java Virtual Machine (JVM) is a major problem for practical deployment. There is currently no free JVM available. Commercially available JVMs for Java 2 ME include IBM's JVM and NSIcom's CrEme. Sun's Personal Java and Insignia's Jeode are not supported anymore and Super Waba is not fully Java compatible. Based on the findings of our pilot study we have selected the .NET compact framework with Visual Basic .NET as the implementation platform for CityInfo. The main reason for this choice was the fact that the .NET compact is pre-installed on current Pocket PC PDAs and ease of deployment was seen as the most important selection criteria.

User Interface (UI) design concept:

A highly intuitive UI design is essential for LBS applications like CityInfo that are intended for general audiences. Key hardware constraints that distinguish mobile applications include:

- *Limited resolution and small display size:* The limited resolution and size of the graphics display of mobile displays is a key constraint. In addition the available color displays are often limited to several thousand colors (compared to "true-color" displays in desktop applications). These factors limit the amount of information that can be conveyed on a single screen. Thus it is often necessary to apply interaction techniques to access all information.
- *Limited processing power:* The limited processing power of mobile devices severely limits the use of interactive real-time animation and the generation of complex graphical displays. Especially the creation of interactive, animated 3D graphics is severely limited because existing mobile devices have no hardware support for 3D rendering.
- *No mouse:* Most graphical user interfaces for desktop devices rely on the use of a 2D pointing device (typically the mouse, but also track-pads, graphic tablets etc.) as the main interaction mechanism. Pointing devices on mobile devices often have significantly lower resolution (e.g. touch-screens) or require the use of additional hand-held components (e.g. pens on PDAs).

On the design side the specific implications of a mobile context of use must also be considered. While desktop applications are used under controlled lighting conditions, mobile devices can be used in situations ranging from total darkness to glaring sun which can have a significant impact on the visibility of graphical presentations that must be considered in the visualization design. Also, since mobile applications are used outside a classical work-environment the level of attention that a user can devote to the application may be limited (e.g. when using the device in parallel to other activities) or user interaction may be interrupted by the user's need to attend to some external events.
For the design of the CityInfo user interface requirements were established through a scenario based requirements elicitation process, based on two usage scenarios. The driving principle behind the design was to use metaphors and mechanisms that are familiar to the largest possible audience, to minimize the amount of interaction required for basic functionality and to enable the integration of arbitrary media elements in the future.

The central element of CityInfo is the (moving) map on which the user's current location and the trail of his past movements is indicated. All additional information and functionality can be accessed from this central element, making the basic functionality of the application instantly accessible to all users that are familiar with maps. The map (Figure 5) is complemented by a menu-bar with graphical buttons providing additional functionality like zooming and route selection. CityInfo provides five different zoom steps (1:2500 to 1:75000) so that the presentation can be adapted according to the specific user requirements from overview to detail maps. The map content for each zoom level is pre-processed during content creation through an adaptive generalisation process. In order to minimize the required interaction the route selection module includes a number of predefined "recommended tours" from which users can select according to interests and preferences. The selected route and points-of-interest (POIs) are integrated into the map display. Additional information for POIs can be accessed by selecting these on the map. This opens a second module, the POI-module (Figure 7). A picture and a short textual description of the chosen POI is displayed for all POIs. Depending on the nature of the POI and the information provided by the content creators additional detailed information can be accessed from this module in different media formats, e.g videos, 3D-views or panorama images. In this module arbitrary presentation formats (e.g. with external viewers) can be integrated for test and evaluation purposes. CityInfo is designed so that the generic functionality can easily be extended. An example for such an extension module is a trip-notebook in which users can document there experience with notes, pictures and trail information while interacting with the system.

Iterative Development Process

Iterative prototyping with frequent testing and evaluation with real users has proven to be a useful method for the exploration of novel and underspecified design spaces where user requirements are difficult to establish. For the CityInfo project we have adopted a workflow based on the ISO standard ISO 13407 [ISO 1999]. ISO 13407 provides an iterative development model that involves users actively in the design process and relies on frequent evaluation, based on current "best practice" in user-centred design. An initial process planning activity makes it possible to adapt the process to application specific requirements. The central design activities in the ISO 13407 cycle are (Figure 3):

1. To analyze and specify the context of use
2. To specify the user and organizational requirements
3. To create design solutions
4. To evaluate the design solutions against the requirements

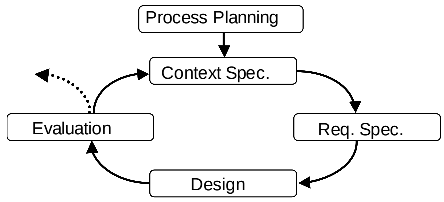

Figure 3: ISO 13407 cycle

The process is iterated until the specified requirements are met. For the purposes of this project we have adopted this method both for the creation of the user interface for end-users, but also as a requirements elicitation method for content development tools, taking the perspective of content creators as "users".

Tools for Multimedia Content Creation

As discussed previously content creation as a whole consists of the creation of media elements by media specific tools in the first step and their integration into a coherent multimedia application in the second. For CityInfo we aim to evaluate a large variety of multimedia elements ranging from text and pictures, over panoramas, maps and videos to interactive 3D models and their corresponding tools for suitability for mobile LBS systems. Key aspects that have to be considered are optimized memory usage and minimized processing power consumption. An additional important aspect is the amount of support for different data-formats. While stand-alone players are available on Pocket PCs for most established multimedia formats the amount of control that can be executed over the presentation varies and is often limited to un-synchronized playback. Also the limitations of the hardware (e.g. no graphics accelerators) preclude the use of many media post-processing techniques like image effects. This makes it necessary to pre-calculate such effects in appropriate tools at the design-time.

1. Text and Pictures (2D raster)

Text and pictures are the basic elements for most multimedia applications. Both are relatively easy and fast to create and enable the presentation of a wide variety of content. For 2D raster graphics capturing, generation and processing a wide variety of tools are available. The main benefit of 2D raster formats in mobile applications is that they require little processing power and can be optimized at design-time for the specific requirements of the application and characteristics of the display. An additional important feature is that support for the standardized graphics file-formats PNG, JPEG, GIF, BMP is provided by VB .NET along with some basic image manipulation functions, making them easily accessible for developers. On the other hand storage requirements are significant and adaptation to other display sizes and formats requires manual interaction for usable results. For our PDA with a screen-size of 480x640 pixels a typical screen-filling raster image has a size of ca. 200KB. In the CityInfo application additional information is provided for all points of interest in text and 2D image formats. Text is used for descriptions and factual information (e.g. historical dates, opening hours etc.). The image of the POI provides basic illustrative content, depending on the nature of the POI.

2. Audio

Audio plays a significant role in many desktop and web multimedia applications. Due to the public and mobile context of a LBS system the use of audio is restricted both due to background noise levels and the fact that understandable audio playback can be inappropriate and embarrassing in public. In these cases the use of earphones is sometimes a viable option. Audio content creation is well supported by a large selection of capturing, generation and processing tools. However, the use of audio in Pocket PC applications is complicated by the lack of integrated support in VB .NET. To playback audio the application has to rely on the Windows Media Player or other external software. The Windows Media Player for Pocket PCs only supports playback of Advanced Streaming Format (.asf), MPEG-1 Layer 3 (.mp3), Windows Media Audio (.wma), Windows Media Player (.wmp) and Windows Media Video file-formats. Audio files in other formats first have to be transcoded into one of these formats. Of these formats .mp3 is supported by most tools. The storage requirements for good quality stereo audio are between 16 KB/s and 20 KB/s. Due to these limitations and restrictions CityInfo uses audio information only as an optional modality that can be selected by the user on demand.

3. Panoramas

During our pilot study panorama images have emerged as a media format that is well suited to LBS applications on PDAs. Together with an interactive scrolling mechanism for changing the viewing direction, panorama images enable the presentation of detailed (partly annotated) landscape images on the portrait format displays of PDAs with reasonable memory requirements. Apart from the interaction mechanism and the specific tools for composing panorama images the same factors as for normal 2D raster images apply.

In CityInfo we employ 360-degree-panoramas at interesting plazas and look-out points. The images are captured with a Nikon D-100 using a wide angle lens (focal distance 14mm). In this setup about 12 pictures (taken vertically) are needed to create a panoramic image. For good quality of the resulting composit it is necessary that the camera rotates around a fixed spot (at best the nodal point of the camera lens) to eliminate parallax and that it stays level. If buildings are very high and close to the camera viewpoint, a second row of images has to be taken with camera rotated upwardly to capture the complete height of the buildings. The single images are stitched together with special panorama software, here we used the freeware Hugin 0.4 beta (Hugin 2005), which is based on the panorama tools from Dersch (2005). The results are panoramic images in jpg file-format, which are scaled and compressed according to the requirements of the application.

4. Vector Graphics

Vector graphics have many attractive features that make them interesting for mobile LBS systems. In a vector graphics system the actual presentation (image) is generated at run-time from a symbolic model description. The advantages of this approach include smaller files size than bitmap graphics, resolution independence that enables continues scaling and arbitrary rotation, the ability to search textual elements in the graphics and the linking of interaction handling to graphical elements. Many vector graphics formats also allow to specify animations. Compared to raster graphics the selection of visual tools for vector graphics creation is much more limited and “programming” in a text editor is too cumbersome for most purposes. While J2ME provides integrated support for vector graphics such functionality is not provided by VB .NET and the .NET compact framework. Therefore, external viewers have to be used. The two main alternatives that are currently available are Macromedia Flash and SVG. Macromedia Flash is a proprietary format that is well established in the web domain and used with a set of corresponding authoring tools. A viewer is available for the Pocket PC platform. The key limitation that prevents the use of Flash for our purposes is the fact that besides the cost of the authoring system a significant yearly license fee applies for the stand alone viewers. The second contenders among vector graphics formats is SVG, an open XML based standard created by the W3C consortium. Some authoring tools for SVG graphics are available and limited support is integrated in vector graphics programs like Adobe Illustrator and Corel DRAW. The problem that arises with the use of SVG is also related to the available viewers. All available viewers are subject to a license fee and the integration with .NET is problematic, although Intesis has announced a version of their eSVG viewer that should work with the .NET compact framework. Our experiments with SVG in the pilot studies have confirmed the benefits of a vector graphics format: SVG content can be created for a large range of communication purposes, incorporating a certain amount of interactivity and animation either using visual authoring tools, programmatic generation or “programming” at the XML code level. However, the deployment problems associated with an external viewer under a commercial license may currently preclude the use of vector graphics in a large range of mobile LBS applications.

5. Maps

Maps are a central element of most LBS systems and form the central element of the CityInfo user interface.

In CityInfo several map layers serve as the basis for all geo-referenced information. The points of interest as well as the planned tour and the user’s current position are presented on top of a map of the city. For the internal presentation and output generation of maps both raster and vector representations can be used. The maps used in CityInfo are prepared as raster maps to make widespread deployment easy. The alternative of SVG vector maps was carefully evaluated since the benefits of vector graphics, e.g. high visual quality, easy scaling and accessibility of individual objects are especially attractive for map presentations. However, as discussed in the previous section on vector graphics the use of SVG was not a viable option under the specific conditions of the CityInfo project.

While raster maps are more easy to handle in the Pocket PC / VB .NET environment they also lead to a number of problems that need to be addressed:

- Direct interaction with map “objects” is not possible. Instead areas of the map image have to be defined as “clickable areas” and associated with user interaction events.
- Problems with handling large data sets. A central limitation of mobile devices is not only the limited memory capacity but also the capabilities of the system software to handle large data sets. In practice graphics elements with a size larger than 1MB are difficult to handle. This makes it impossible to represent the whole area of the city of Hanover as a single map image. Instead the map has to be separated into smaller elements. To achieve this the map is dissected into several map segments from which the actual presentation is composed. While panning the coverage of the current map sheet is checked and new map segments are loaded for presentation if required.
- Support for multiple scales. The small display of mobile devices necessitates the use of different map scales ranging from overview to local detail. Our pilot-study has shown that support for the following map scales is desirable for a city navigation and information application like CityInfo: 1:2.500, 1:5.000, 1:10.000, 1:25.000 and 1:50.000. Scaling of map images is not a viable option due to unacceptable quality. Therefore, if raster maps are to be used a suitable map series has to be produced in advance and stored on the mobile device, with the associated memory requirements.

The required map images for CityInfo covering buildings, roads, railroads and water as well as separate layers for the points of interest and the sightseeing routes generated from existing data sets. As the basis for our maps we use ATKIS-data from the mapping agency of Lower Saxonia as well as cadastral data, since buildings are not included in the ATKIS data definition. ATKIS is the German topographic cartographic spatial database and presently contains more than 60 different feature types for the whole area of Germany in the scale of 1:25.000 (beside this scale there are further levels of data aggregation in the scales 1:200.000 and 1:1.000.000 which were not used in this work).

The raw vector data first has to be symbolised and generalised for the planned scales. For the symbolisation as well as for the generalisation the minimum dimension of ordinary paper maps has to be factorised by two to four, as the resolution of the mobile devices is only 200 dpi. Compared to this paper maps have a resolution up to 1.000 dpi and more. The buildings had to be simplified and amalgamated for the scales smaller than 1:5k, using the software system CHANGE, developed at the University of Hannover. The results can be seen in Figure 4. Furthermore the buildings as well as the linear objects had to be re-located as after the symbolisation of the linear objects some of the buildings can be located in undesirable places, e.g. on the road. A second software system called PUSH displaces the buildings and rearranges the roads as long as a solution can be found where no collision between these two datasets occurs. More details about the software system PUSH and the displacement algorithm can be found in Sester (2000). All map images used in the application are geo referenced to match the map with the GPS position of the user.

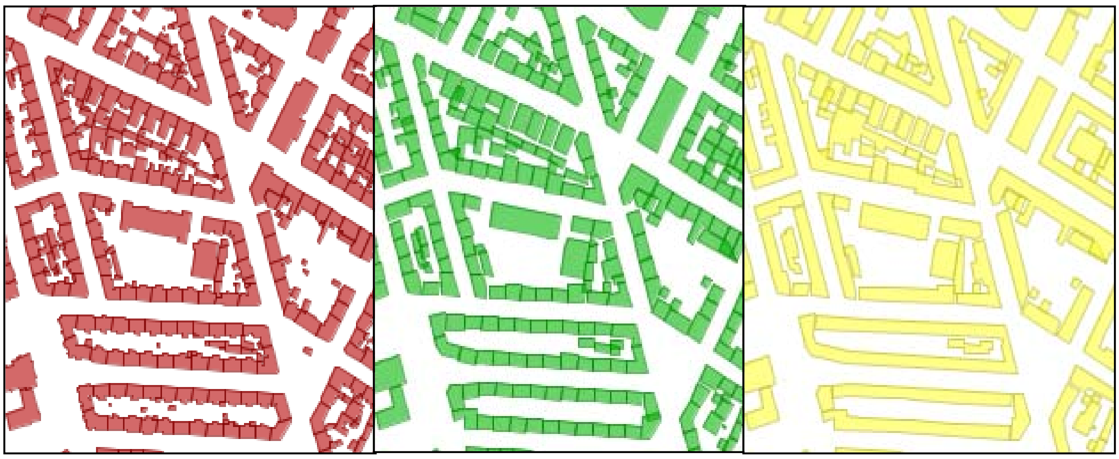

Figure 4. Generalising buildings: Original data (left), simplified data for scale 1:10k (middle), simplified and merged data for scale 1:25k (right). Data from “Katasteramt Hannover”.

6. Video

Video is a major component in most multimedia systems and the development of “interactive” video systems can be seen as one of the starting points towards the multimedia systems of today. Video as a time-based media format shares many of the properties of audio discussed above. Due to the digitalisation of television technology (DVB and DVD), the widespread use of digital video cameras and video capabilities integrated in today’s mobile phones and digital cameras a wide range of digital video capturing and manipulation tools has become available that addresses the complete spectrum from amateur to professional video editing. For desktop and web applications a wide variety of video codecs (Compressor-Decompressor) are available that offer different tradeoffs between quality, size of the data stream and processor performance requirements. As with audio the use of video in Pocket PC applications is complicated by the lack of integrated support in VB .NET. While J2ME offers limited video support through extension packages the Pocket PC/ .NET combination relies on the Windows Media Player or other external players for video presentations. Since the Windows Media Player for Pocket PCs only supports playback of Advanced Streaming Format (.asf), Windows Media Player (.wmp) and Windows Media Video video file-formats other formats first have to be transcoded into one of these formats or an external player has to be used. A popular example for such a third-party player is the PocketTV player by MpegTV that supports the widely used mpeg video format and is available in a free classic version. However, if ease of deployment is a requirement (as in CityInfo) developers are restricted to the formats supported by the Windows Media Player. For a 320x240 video memory requirements are around 50 KB/s for poor quality (15 frames per second) and 100 KB/s for low quality (30 frames per second) with virtually no upper limit for higher levels of quality and resolution.

7. 3D Graphics

3D graphics is the collective name for several cues that can – individually or in concert – create the impression of depth in images. In practice the monocular depth cues of perspective and shading are typically identified with “3D graphics”. The images in 3D graphics presentations are rendered at run-time from geometric model descriptions, similar in this respect to vector graphics. Since the rendering process approximates the function of a camera both stereoscopic and non-stereoscopic images can be created from the same models. In desktop PCs specialised hardware in graphics card is used to accelerate the graphics rendering from the 3D models provided. Current PDAs do not yet support hardware accelerated rendering, which severely limits the complexity of the 3D models that can be handled. However, the first hardware accelerators for mobile devices are announced based on the new Open GL ES standard, driven largely by the fast development of the smartphone market. The central prerequisite for 3D graphics rendering is the availability of adequate 3D models. A variety of well-known 3D modeling and animation tools are available for 3D model creation. Due to the limitations of current PDAs it is essential to minimize the complexity of the 3D models during the modelling process. An alternative is the acquisition of spatial 3D data from real environments, e.g. using laser scanning. While established tools can be used for the initial acquisition process the reduction of the resulting 3D models to a level of complexity that can be handled by current PDAs is difficult and typically requires manual intervention. For the CityInfo project we use the well-known 3D modeling and animation tool 3D Max. 3D objects for which laser scanner data is available are manually remodelled in 3D Max to derive models with adequate complexity.

The second prerequisite for 3D graphics is a renderer with a programming interface. In the desktop domain Open GL and Microsoft’s Direct 3D have emerged as the standards for 3D graphics rendering. Both are immediate-mode graphics libraries that provide direct access to the capabilities of today’s graphics cards and work on the level of polygons. Several high-level graphics libraries are available that build on these to provide programmers with a more intuitive way of generating 3D graphics from object models. For PDAs the situation is different. No standardized support for 3D graphics is available from the Pocket PC operating system. In fact the so called game-API by Microsoft only provides programmers with a pointer to the frame-buffer of the mobile device, so that individual pixels can be changed manually, a situation that is clearly un-acceptable for content-orientet creation of LBS applications. One possible solution is to use a software implementation of Open GL that provides a basic set of 3D rendering functionaly to implement 3D graphics. A second option is to use a commercially available 3D viewer with an integrated software renderer. The shortcoming of the first approach is that the 3D capabilities are still at a very basic level (display of polygons). The use of a commercial 3D viewer based on the VRML (Virtual Reality Modelling Language) or its follow up X3D is much simpler and enables content creators to work on the object level, using 3D models created by authoring tools. The main restriction with VRML viewers like Pocket Cortona is the lack of programmability. The functionality is largely limited to displaying of 3D models. External control of the 3D content for advanced interaction and animation functions is not possible. Both approaches require additional software packages, creating problems for deployment.

3D graphics on mobile devices is an area of fast development. The emerging market of games on smartphones has initiated a number of interesting developments in the recent past. The first is the creation of the Open GL ES (embedded systems) standard, a simplified version of the general Open GL standard that takes the specific requirements of mobile devices into account. A number of hardware accelerators for smartphones that support Open GL ES are currently under development which will eventually enable the use of much more detailed 3D models. For LBS programmers the specification of the Java extension JSR-184 "Mobile3D Graphics" is an interesting development. It specifies both a low level immediate mode programming interface that exposes the hardware oriented functionality of Open GL ES and a high level retained mode rendering system based on the object oriented scene graph paradigm.

For the CityInfo system with Pocket PC PDAs as the target platform we use an external VRML viewer to provide access to 3D models of selected POIs.

Examples of Content in the CityInfo Prototype

As discussed previously the development of CityInfo continues as part of an iterative process in order to evaluate existing development tools, to study the behavior of content authors and to elicitate their requirements. To study a large population of content authors the approach was used in a one year student project in which a wide variety of content for LBS applications was created by several teams. The following screen shots (Figures 5-8) illustrate various content types created as part of this process and shows the concrete implementation of the user interface concept introduced previously.

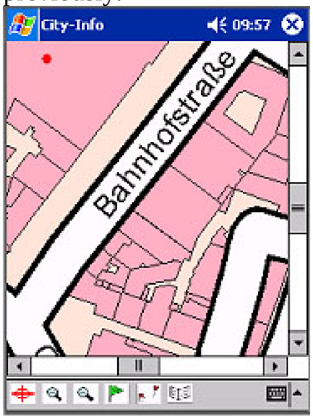

Figure 5: Map Module

Figure 6: Route Module

Figure 7: POI Module

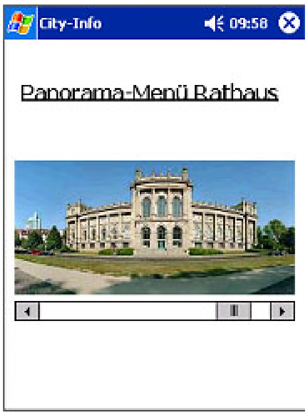

Figure 8: Panorama Module

Results and Outlook

In addition to the concrete implementation of the user interface and content of a city information system shown in the previous section the CityInfo project has also led to a number of interesting insights regarding possible platforms for LBS applications, development environments and existing tools that are relevant in a larger context.

Hardware and software platforms: PDAs still have some advantages over smartphones in raw processing power, memory size and extensibility with external components like GPS receivers. However, hard- and software development in the smartphone domain is much faster than for Pocket PC PDAs, that differ little from the first generation of Pocket PCs (e.g. Compaq's iPAQ H3650) introduced in 2000. Smartphones have already surpassed PDAs in terms of sales figures, and the announcement of hardware support for 3D graphics and GPS positioning in future smartphones makes these a much more attractive platform in the near future. On the software side both the Java and VB .NET approach have their benefits and shortcomings and rely on external extensions for advanced multimedia capabilities. The development of standardized multimedia API's for Java (e.g. JSR-135 Mobile Media API, JSR-184 Mobile 3D Graphics API, JSR-226 Scalable 2D Vector Graphics API and JSR-234 Advanced Multimedia Supplements) and their integration into smartphones using the Symbian operating system are additional aspects that make smartphones an attractive platform. However, the need for a SIM card in current smartphones can be problematic, e.g. for use as part of a student project.

Development environments: Both Microsoft's Visual Studio (using VB .NET or C#) and the various Java development environments are viable approaches to LBS development. A central problem that has emerged with the VB .NET approach in the CityInfo project is the difficulty of accessing appropriate documentation. While the .NET compact framework supports only a fraction of the functionality of the complete .NET framework, its documentation is embedded in the same documentation by marking individual elements as "supported in the compact framework". Finding the required and applicable information in the voluminous .NET documentation is problematic for non-experts like students and content developers that have no interest in the deeper internals of the .NET framework on the desktop. An additional problem arises due to the limitations of the .NET compact framework. As many commonly required functions (e.g. reading serial ports or bluetooth for access to GPS data) are not supported in the compact framework

direct access to the underlying Win32 API through the clumsy P/Invoke mechanism is often required. While the quality of the documentation is not necessarily better the documentation of open standards like the set of Java APIs is much more accessible for casual developers.
Existing tools: One of the aims of the CityInfo project was to evaluate existing tools for multimedia production with regards to their suitability for production of LBS content. Overall the experience has been quite positive as described for the individual modalities. Few problems arise with the adaptation of existing tools for individual media creation as many requirements are similar to multimedia development for the web (e.g. low-memory and bandwidth requirements) and therefore well supported. Interesting additions could include specific tools for 3D modeling with low-polygon numbers and better support for media synchronization. One shortcoming we have identified is the lack of appropriate authoring tools to combine different media into a coherent and interactive multimedia presentation. Existing tools lack required support for LBS functionality like GPS based positioning and spatial data access and for proposed open standards like SMIL, MPEG-4,7 and 21 software support in the form of appropriate players is still missing.
For the future it will therefore be necessary to address the specific authoring requirements of LBS applications and support them through suitable extensions of standards to include location based interaction and content access.
Our next step in the CityInfo project will therefore be a thorough requirements analysis with regards to high-level authoring tools and a specification of the base functionality that should be provided for the creation of LBS. In order to recruit a significant number of developers for theses time consuming experiments we plan to continue our approach and observe the use of the system in student projects within a university course on LBS development.

References

- Baus, J. Krüger, A. and W. Wahlster 2002: “A Resource-Adaptive Mobile Navigation System”, in Proc. of IUI’02, 7th international conference on Intelligent user interfaces, San Francisco, California, USA, pp. 15-22.
- Baus, J., Cheverst, K. and C. Kray 2005: “A Survey of Map-based Mobile Guides”, in: Map-based Mobile Services Theories, Methods and Implementations, Springer, pp.197-213.
- Dersch, H. 2005: Panorama Tools, http://www.path.unimelb.edu.au/~dersch/, visited: 18.04.05.
- Dunlop and Brewster 2001: Proc. of Mobile HCI 2001: 3rd Int. Workshop on Human Computer Interaction with Mobile Devices, IHM-HCI 2001 Lille, France, September 2001
- Elias, B., Hampe, M. and Sester, M. 2005: “Adaptive Visualisation of Landmarks using an MRDB”, in: Map-based Mobile Services Theories, Methods and Implementations, Springer, pp. 75-88.
- Gartner, G, Frank, A. and Retscher. G. 2004: “Pedestrian Navigation System for mixed indoor/outdoor environments”; in: "Location Based Services & TeleCartography", 66 (2004), pp. 161 - 167
- Gartner, G. and Uhlirz, S., 2001: “Cartographic Concepts for Realizing a Location Based UMTS Service: Vienna City Guide Lol@”, in: Proc. 20. International Cartographic Conference, 06.-10.08.2001, Bejing
- Geiger, C.; Paelke, V.; Reimann, C. and Stoecklein, J. 2002: “Interaktive Visuelle Inhalte auf Mobilen Endgeräten” in: Proc. Simulation und Visualisierung 2002, Magdeburg, March 2002
- Heidmann, F., Hermann, F. and M. Peissner, 2003: “Interactive Maps on Mobile, Location-Based Systems: Design Solutions and Usability Testing”, in: Proc. 21st International Cartographic Conference (ICC) Durban, pp. 1299-1305.
- Hugin 2005: Hugin – Panorama Tools GUI, http://hugin.sourceforge.net/, visited: 18.04.05.
- ISO 1999: “Human-centred design processes for interactive systems”, ISO Standard 13407
- Johnson, 1998: Proc. First Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1, University of Glasgow, Scotland
- Kolbe, T. 2004: “Augmented panorama and videos for pedestrian navigation”, in: Proc. 2nd Symposium on Location Based Services and TeleCartography 2004, Geowissenschaftliche Mitteilungen, TU Wien.
- Pospischil, G., Umlauft, M. and E. Michlmayr, 2002: „Designing LoL@, a Mobile Tourist Guide for UMTS”, in: F. Paterno (Ed.): Mobile HCI 2002, LNCS 2411, Springer Verlag, pp.140-154.
- Rosson , M.; Kellogg, W and Maass, S. 1988: “The designer as user: building requirements for design tools from design practice”, in: Communications of the ACM, Vol. 31 , No. 11, pp. 1288 - 1298
- Sarjakoski, T., Sarjakoski, L. T., Lehto, L., Sester, M., Illert, A., Nissen, F., Rystedt, R. and R. Ruotsalainen , 2002: “Geospatial Info-mobility Services - A Challenge for National Mapping Agencies” in: Proc. Joint Int. Symposium on "GeoSpatial Theory, Processing and Applications" (ISPRS/Commission IV, SDH2002), Ottawa, Canada, 5 p
- Schilling, A., Coors, V. and K. Laakso 2005: “Dynamic 3D Maps for Mobile Tourism Applications”, in: Meng, Liqiu; Zipf, Alexander; Reichenbacher, Tumasch (Eds.): Map-based Mobile Services Theories, Methods and Implementations, Springer, pp. 233-244.
- Sester, M 2000.: “Generalization based on Least Squares Adjustment”, in: International Archives of Photogrammetry and Remote Sensing, Vol. 33, ISPRS, Amsterdam.
- WMCSA, 2002: Proc. IEEE Workshop on Mobile Computing Systems and Applications