

Mobile Guide For The Enterprise

Christian SCHMITT

Fraunhofer FIT, Sankt-Augustin, Germany, email : christian.schmitt@fit.fraunhofer.de

Abstract. This paper describes a mobile guide running on the Pocket PC platform that presents information about our research department, including general information, running projects, employees, events and more. It uses a combination of Wireless LAN indoor positioning and Scalable Vector Graphics (SVG) based map visualization to provide visitors with indoor navigation on the FIT's floor. Wireless LAN positioning is an easy to setup solution providing reasonable accuracy whereas SVG offers numerous advantages for navigation including small file size, high resolution zooming, animation and scripting.

1 Introduction

FIT has a long history in developing mobile guides for museums, fairs, symposiums. In the HIPS project [1] we developed a mobile guide for our Castle in Birlinghoven. Infrared emitters attached to the Castle's pictures were used to locate the user and generate personalized tours or navigation hints. In 2003 we developed a mobile guide for our Open Day in April. The visitors were mostly impressed by the user tracking and the new opportunities offered by indoor localization and visualisation of objects of interest.

In this paper we will sketch an overview of existing localization technologies for indoor navigation. We will describe in more detail the selected solution which is based on Wireless LAN. This paper will have a close look at positioning and mobile navigation and will present a solution based on the promising web standard for vector graphics, SVG.

2 Indoor Positioning overview

There are numerous technologies available for user tracking. Infrared (IR) ([6], [7]) is cheap and available on a broad palette of devices ranging from PDAs, mobile phones to notebooks and remote controls. Unfortunately it is painful and hard to setup and maintain such a positioning system (lot of emitters are required because of the directed and small coverage of IR power batteries have to be replaced from time to time). Finally IR won't work if there is no line of sight between the device and the emitter (which often happens in overcrowded museums or fairs) or where unfavourable light conditions exceed (e.g. strong external lighting on a sunny day).

Radio Frequency (RF) based systems ([2],[3],[4],[5]) are also widely used because they overcome many of the disadvantages of IR. The signals are able to pass through walls and obstacles, thus having a better coverage and therefore requiring less RF receivers to setup a whole building. Drawbacks are multiple path effects and signal attenuation that require additional and often intensive processing to filter out noise. In addition, the frequency ranges are usually licensed and restricted to license free ranges available. Systems with high precision (~10-30 cm) measure the time of propagation between the device and the receiver and require a synchronization of the receivers (base stations). The disadvantage of those systems is their price. Systems relying on the signal strength measurements are less complicated and therefore cheaper, and less precision (~1-3 m).

There is also a range of optic systems based on image processing. The idea of those systems is to mimic the capabilities of the human eye and brain. Through edge detection or object identification

from the raw image and comparison with a image-database it is possible to compute the distance to well-known objects and hence the absolute position of the camera. But image processing is a very CPU-intensive task and requires a very precise camera, making those systems very expensive (with high-precision cameras and dedicated server for image processing) or not precise enough (with low-budget cameras and mobile image processing).

3 WLAN Positioning with Ekahau

The Finnish company Ekahau [8] has developed a 100% software based positioning system that can use the existing WLAN infrastructure for positioning, making this solution very cheap and flexible.

WLAN Positioning uses the signal strength of the access points in the neighbourhood to calculate the position. There are two main approaches to determine the position. The empirical model is based on comparing the current signal strength readings to a database of pre-recorded measurements. A calibration phase creating a radio map has to be performed, before the system can be used. The second approach is called the propagation model and is based on mathematical equations describing the degradation of the signal strength of a radio wave in space. Using these propagation equations an approximate distance from the WLAN card to the access point can be computed and through triangulation an absolute position. The Ekahau solution is based on the empirical model.

The position information (X-Y coordinates, heading and speed) about any connected device can be queried from the Ekahau Positioning Engine (responsible for gathering the fingerprints from the mobile devices and computing their position) using either the Ekahau SDK (Java 1.4 support needed) or using a socket-based proprietary protocol called Yax (for devices without 1.4 support, like most PDAs and mobile phones). During development time Yax was not yet available, therefore a similar protocol had to be designed to retrieve the position of the PDA. Mobile devices use this protocol to interact with a middleware server connected to the Positioning Engine through the SDK.

4 Mobile Navigation with SVG

The next step after being able to retrieve the current position of the device was to display that position on the PDA screen. We were interested in implementing the functionality for information browsing into the map. For example, clicking on the name of an employee should popup a window with the employee's personal details. Points of interests (POI) like paintings or the coffee machine should also be accessible by clicking them.

4.1 The good combination

Our first implementation was a small personal java application rendering a raster bitmap and supporting zoom in/out, manual or automatic scrolling on the current position. Navigation was supported by using an overlay model of interconnected nodes representing possible destinations and implementing a shortest path algorithm. It resulted into a "green mile" showing the path leading to the desired destination. This solution suffered from some drawbacks. First, raster bitmaps do not render nicely when zooming in/out. Second they use a lot of space in memory in the already resource limited PDAs. Furthermore, Java applications are generally slow to load on PDAs, do not have rich user interfaces and are not using optimally the limited resources of PDAs.

Using these lessons we started a more ambitious project based on Scalable Vector Graphics (SVG) for the map and Visual C++ for Windows CE (Pocket PC 2002). By using the C++ we could improve the overall performance while getting more control over the device (support for hardware buttons, Internet Explorer and Media Player ActiveX controls). The user interface could be designed more user-friendly because operating system widgets could be used, that make sure that the user can interact in a known interface without risking discomfort and disorientation.

Vector graphics have a smaller footprint than bitmaps and will render a map at any scale without any loss of detail. In 1999 the W3C started working on a new, open, xml-based vector graphics standard for the web. SVG rapidly gained momentum and established itself as the perfect match for visualizing maps. Indeed, SVG maps are small in size, provide high-resolution zooming and panning, gradients, animation, symbols, linking to other web pages and last but not least scripting functionality.

The W3C released two mobile profiles of SVG 1.1 in January 2003: SVG Basic for PDAs and SVG Tiny for mobile phones [9]. There are only two C++ viewers implementing SVG Basic, BitFlash [11] and eSVG [12]. BitFlash is faster than eSVG but eSVG provides a mechanism for the embedding application to be notified of the user's interactions with the map (mouse click, zooming, etc). This feature is helpful in context-aware systems that have to learn the context from the user's interaction and the reason why we favoured eSVG.

4.2 The SVG Map

The map has been designed using BitFlash's Brilliance SVG editor which generates SVG Basic code that will run for sure with eSVG. Other more powerful editors like Illustrator are outputting heavy SVG code too hard to use on mobile devices.

Here is an extract of the SVG map:

```
<svg width="486.0px" height="1294.0px" viewBox=" 0 0 486 1294" onload="OnLoadEvent(evt)">
<script type="text/ecmascript" xlink:href="LocateOnEverySecond.js"/>
<g id="grundriss">
<rect x="0" y="0" onmouseover="updateLastId(evt)" fill="none" width="116" style="stroke: rgb(0, 0, 0);" id="room101" height="66"/>
<rect x="0" y="66" fill="none" width="116" style="stroke: rgb(0, 0, 0);" id="room102" height="66"/>
...
<rect x="324.0" y="1181.0" onmouseover="updateLastId(evt)" onclick="2" width="15" style="stroke: rgb(255, 0, 0);fill: rgb(255, 0, 0);" height="34.0" id="armor"/>
<rect x="324.0" y="1240.0" onmouseover="updateLastId(evt)" onclick="2" width="15" style="stroke: rgb(255, 0, 0);fill: rgb(255, 0, 0);" height="34.0" id="bathseba"/>
</g>
<g id="names">
<text x="41" font-size="12" y="22" onmouseover="updateLastId(evt)" fill="#000000" onclick="1" font-family="Arial" id="seeling">Seeling
</text>
<text x="45" font-size="12" y="43" onmouseover="updateLastId(evt)" fill="#000000" onclick="1" font-family="Arial" id="becks">Becks
</text>
...
<text x="202" font-size="12" y="620" onmouseover="updateLastId(evt)" fill="#000000" onclick="4" font-family="Arial" id="konferenzraum">Konferenzraum
</text>
</g>
```

It is always a good practice to assign all elements an id. The SVG document references a ECMA-Script file that contains the logic responsible for adapting the display to the context (position). When the document is loaded the OnLoadEvent(evt) function is started. It will setup the communication channel between the ActiveX control and the main application (used to pass by new location coordinates or tour information) and start a timer mechanism to refresh the current position every second. This communication channel allows exchanging properties (String peers).

Let us examine how the information browsing has been realized. By information browsing is meant the triggering of a new window to displays information about the clicked element. This could have been done in SVG using scripting but we wanted to use the dialogs of the main application instead. For that purpose, eSVG provides 32 callback functions that can be caught by the main application. To do so, the elements that should generate such a callback function have to use the number of the callback function as part of a mouse event's attribute. Every time the user moves the pen over an element the id of that element, let say n, is saved in a script variable by the `updateLastID(evt)` function. Immediately after releasing the pen a mouse click is generated that calls the `OnCallback[n]Activexesvgctrl2` function of the main application's dialog that embeds the map control. Here are some of the implemented callback functions. The first handles a click on an employee name and opens a dialog window with the corresponding personal information. The second starts an animation of the

```
//callback function for employee data
void CMapDlg::OnCallback1Activexesvgctrl2(LPDISPATCH evt) {
    CString lastID=m_Obj.getProperty(L"lastObjectID");
    CExhibitorItem dummyItem;
    dummyItem.lastname=lastID;
    CExhibitorItem* item=theApp.EmployeesList.GetByName(dummyItem);
    if (item==NULL) return;
    CExhibitorDetail exDetail;
    exDetail.idx = 0;
    exDetail.SetExhibitorItem(item);
    exDetail.DoModal();
}
//callback function for paintings
void CMapDlg::OnCallback2Activexesvgctrl2(LPDISPATCH evt) {
    CString lastID=m_Obj.getProperty(L"lastObjectID");
    Paintings dlg;
    if (lastID.CompareNoCase(_T("armor"))==0) {
        CAnimation animArmor;
        animArmor.LoadPageURL(L"file://\\OpenDoorData\\Paintings\\form\\form1.html");
        animArmor.DoModal();
    }
}
```

For the display of the current position, a set of basic shapes (circle and lines) are combined into a human-like form. Their position is updated every second by the script using a timer. In addition the position is always centred in the middle of the screen using appropriate *translate* value for the *transform* attribute

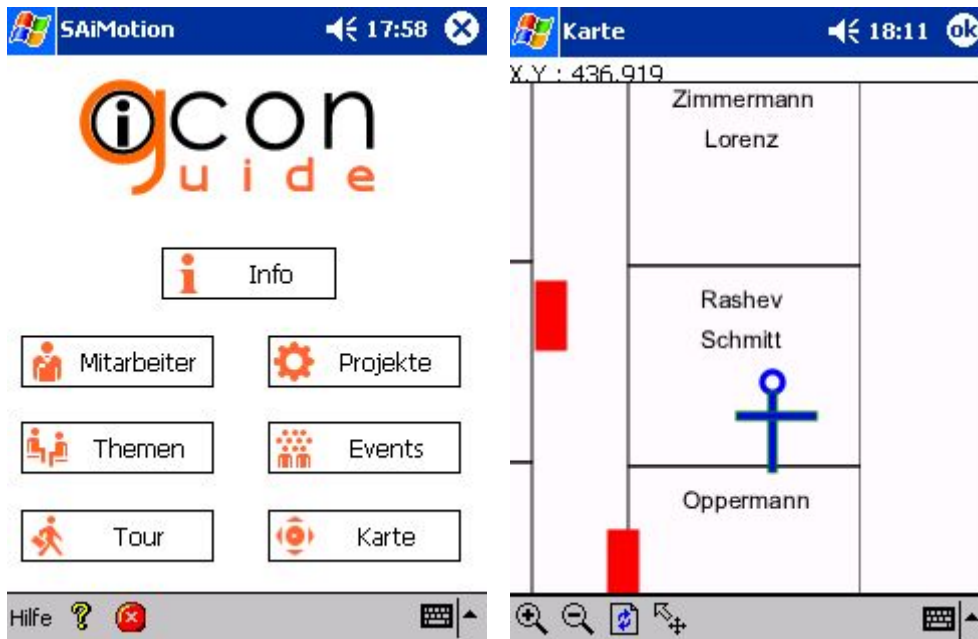


Fig 1: The ICON Guide start page (left) and the display of the user's current position (right)

4.3 The issues

During the development of the mobile guide a number of issues did arise:

- Slow SVG performance.
- Scripting or main application logic (in addition to server logic)

The overall performance of the eSVG viewer was rather average. The main subject of concern for navigation on a map is the panning mode. After panning the map there is a significant delay in repainting the blank area. It leads to a dramatic drop in the general acceptance of SVG. Due to memory optimizations the whole map cannot be stored in memory as a bitmap, hence it is not possible to achieve fast scrolling of the map. The BitFlash viewer proved to be faster than eSVG and in particular didn't have problems with blank areas. Therefore we plan to use the BitFlash SDK to gain full access to the SVG map functionality from the main application.

There is usually a trade-off between server and client logic when designing a cartographic or web mapping application. The typical way of viewing SVG content is to embed the SVG content into the browser. The server is responsible for generating personalized content, while clients rely on scripting for interaction and further personalization. If the SVG content is embedded into your own application you can take advantage of the application language capabilities for further interaction on the client side. For example, the current position of the user is read from the socket connection by the main application. The main application forwards the position coordinates to the map control and updates the transform attributes of the outermost SVG element in order to focus the map on the current position. It is an important research issue to decide how to distribute functionalities between the different parties and what the decision criteria (performance, flexibility and programming power) are.

5 Conclusion

There are a lot of indoor positioning technologies available. In cases where an average precision is sufficient (1-3 meters) and a WLAN infrastructure is already available, the best match is WLAN positioning because it will not require additional hardware and the only costs are the license for the localisation software.

We are convinced that the numerous advantages of vector graphics particularly in SVG abound and SVG will represent the leading technology in the field of mobile navigation as soon as the SVG support on mobile devices will be good enough. The same opinion is shared by the 3rd Generation Partnership Project (3GPP) who has selected Mobile SVG as the mandatory vector graphics media format in MMS (Multimedia Messaging Service). Indeed, companies like BitFlash and ZOOOMON have recently announced deals with major cell phone manufacturers to integrate their SVG viewer into mobile phones.

Future implementation work will include the integration of additional positioning systems (GPS, IR, camera systems) into a campus guide for the Fraunhofer site at Schloss Birlinghoven and the evaluation of the BitFlash SVG SDK. Research work will concentrate on adding context-aware information to SVG in order to reach the next generation of Location Based Services.

References

- [1] R. Oppermann and M. Specht, "A Context-sensitive Nomadic Information System as an Exhibition Guide" in Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, HUC 2000, Bristol, UK, September 25-27, 2000, pp. 127 - 142.
- [2] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system", in Proceedings of IEEE Infocom 2000, March 2000, volume 2, pp. 775-784
- [3] R. Want et al., "The Active Badge Location System", ACM Transactions on Information Systems, January 1992, pp. 91-92.
- [4] A. Ward, A. Jones, A. Hopper. "A New Location Technique for the Active Office", in Proceedings of Mobicom '97, September 1997.
- [5] N. Priyanatha, A. Chakraborty, and H. Balakrishnan, "The Cricket location support system," in Proceedings of MOBICOM, Aug. 2000, pp. 32-43.
- [6] A. Butz, J. Baus, and A. Krüger. „Augmenting buildings with infrared information”, in Proceedings of the International Symposium on Augmented Reality, 2000
- [7] Eyeled GmbH, <http://www.eyeled.de>
- [8] Ekahau, <http://www.ekahau.com>
- [9] Mobile SVG Profiles, <http://www.w3.org/TR/SVGMobile/>
- [10] S.D. Feller and al, "Tracking and imaging humans on heterogeneous infrared sensor array for tactical applications", SPIE Aerosense 2002
- [11] BitFlash, <http://www.bitflash.com>
- [12] eSVG, <http://www.embedding.net/eSVG/>