CERN Building and Map Interface via WWW and on the Palm

Mark Dönszelmann[1]

[1] CERN, CH-1211 Geneva 23, Switzerland

Abstract

The buildings at CERN are chrono-chaotically numbered. They are therefore sometimes hard to find without a map and index.

A WWW browser can be used to retrieve the location of a building by name or number. A set of electronic maps containing the Meyrin, Prevessin, LEP and SPS sites are available in PostScript format. These maps are used to build a database of the locations of the buildings which together with the maps are accessed by some CGI scripts to provide the browser with a set of GIF bitmap images. These images show the full CERN site, the site of the building and an enlargement of the exact location of the building. A click on the image can show any location at CERN or change the enlargement factor.

A Palm application can be used to access the same maps. In this application the maps are stored on the Palm and are drawn on request in real time. The user is thus able to lookup buildings and to interactively scale and translate the resulting picture. The limited memory and CPU resources which are available in the Palm are used to store the maps and building coordinates and to calculate the picture. The application is written in Java to provide portability in future to other hand held devices. It runs on the recently released kvm, an implementation of the Java Micro Edition on top of PalmOS.

Keywords: Maps; Buildings; PERL; CGI; Java; kvm; Palm; PalmOS;

1 Buildings and Maps

CERN has more than ten sites containing over 500 buildings. These buildings were often numbered by function, sometimes by site, but most of the time they got their number historically[1]. To find your way around the chrono-chaotically numbered buildings at CERN you need a map and an index. Apart from hardcopies you could already have online access via the web[1] using a browser since four years. Recently offline access to the maps has been added, using a PDA (Personal Digital Assistant)[2], such as a Palm[2]. Both Web and Palm interfaces use several maps, drawn up in PostScript[3] using Adobe Illustrator[4], as their source of information. Building locations and detailed maps are all derived from a single source. The following discusses both interfaces in more detail.

2 The Web Interface

The Web interface is based on a standard Web browser able to show images in GIF format and a set of CGI-scripts[5] written in PERL[6]. A web form allows the user to lookup a building by number, using the “Building URL”, see Table 1. The building location is searched for in a coordinate database, using a CGI-script. This database is extracted offline from the original PostScript map The result of the lookup is an HTML file containing a reference in the form of a “Map URL”, see Table 1, which contains enough information for the web server to produce a map. The browser automatically requests this “Map URL” from the server, which invokes.another

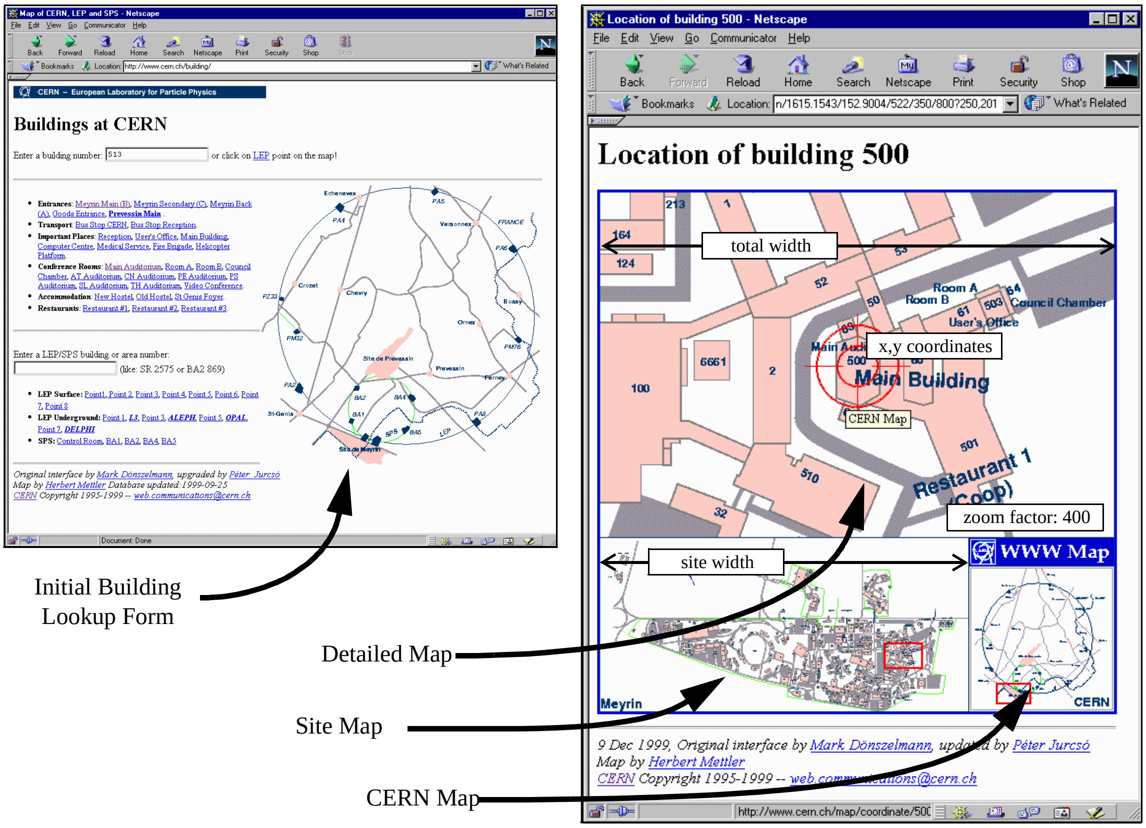
[1] Web interface available at: **http://www.cern.ch/building**

[2] PDA interface available at: **http://palm.cern.ch/**

CGI-script to generate a GIF showing the location of the building ona detailed map, the location of the detailed map on a site map, and the location of the site map on the CERN map, see Figure 1.

Table 1: The 3 URL formats handled by 3 CGI-scripts to do a lookup and to produce a map.

	URL (with prefix http://www.cern.ch/map/	**Result**
Building URL	building?bno=500	HTML file with reference to a“Map URL”
Map URL	map/meyrin/1615.1543/152.9004/522/350/400.gif map name x-coordinate y-coordinate total width site width zoom factor	Generated GIF with three maps
Coordinate URL	coordinate/500/meyrin/1615.1543/152.9004/522/350/400?205,201 building number map name x-coordinate y-coordinate total width site width zoom factor click x-coordinate click y-coordinate	HTML file with reference to a “Map URL”



Map of CERN, LEP and SPS - Netscape

Location: http://www.cern.ch/building/

CERN – European Laboratory for Particle Physics

Buildings at CERN

Enter a building number: 513 or click on LEP point on the map!

- **Entrances**: Meyrin Main (B), Meyrin Secondary (C), Meyrin Back (A), Goods Entrance, **Prevessin Main** .
- **Transport**: Bus Stop CERN, Bus Stop Reception.
- **Important Places**: Reception, User's Office, Main Building, Computer Centre, Medical Service, Fire Brigade, Helicopter Platform.
- **Conference Rooms**: Main Auditorium, Room A, Room B, Council Chamber, AT Auditorium, CN Auditorium, PE Auditorium, PS Auditorium, SL Auditorium, TH Auditorium, Video Conference.
- **Accommodation**: New Hostel, Old Hostel, St Genis Foyer.
- **Restaurants**: Restaurant #1, Restaurant #2, Restaurant #3.

Enter a LEP/SPS building or area number:

(like: SR 2575 or BA2 869)

- **LEP Surface:** Point1, Point 2, Point 3, Point 4, Point 5, Point 6, Point 7, Point 8
- **LEP Underground:** Point 1, ***L3***, Point 3, ***ALEPH***, Point 5, ***OPAL***, Point 7, ***DELPHI***
- **SPS:** Control Room, BA1, BA2, BA4, BA5

Original interface by Mark Dönszelmann, upgraded by Péter Jurcsó
Map by Herbert Mettler Database updated: 1999-09-25
CERN Copyright 1995-1999 -- web.communications@cern.ch

Document: Done

Location of building 500 - Netscape

Location: n/1615.1543/152.9004/522/350/800?250,201

Location of building 500

9 Dec 1999, Original interface by Mark Dönszelmann, updated by Péter Jurcsó
Map by Herbert Mettler
CERN Copyright 1995-1999 -- web.communications@cern.ch

http://www.cern.ch/map/coordinate/50(

Figure 1: Building interface via the Web. On the left the initial building lookup form, with an overview map of CERN, on the right the result of the lookup, showing detailed information of building 500, its location on the Meyrin site and the location of the Meyrin site within CERN.

A third CGI-script handles the “Coordinate URL” when the user clicks around on the map. The result of this Coordinate URL is again an HTML file with a reference to a new Map URL, which will produce a map with more or less detail, depending on the area where the user clicked.

The generation of the map image GIF is done by using the original PostScript map file and using GhostScript[7] to zoom in on the detail, translate and crop the resulting picture. A special PostScript converter file is used to produce a bitmap file compatible with the NetPBM package[8]. The NetPBM package is used to add targets and squares on the three maps, to glue the three maps together and to convert them finally into one GIF file. The dataflow of the web interface is shown in Figure 2. The on-the-fly production of a GIF currently takes a few seconds.

The database is produced offline with a converter program. It scans multiple PostScript map files for text entries and builds a database file to lookup text against maps and coordinates. GIFs for buildings can be produced offline and cached for fast retrieval. Each GIF is about 12 Kbytes.

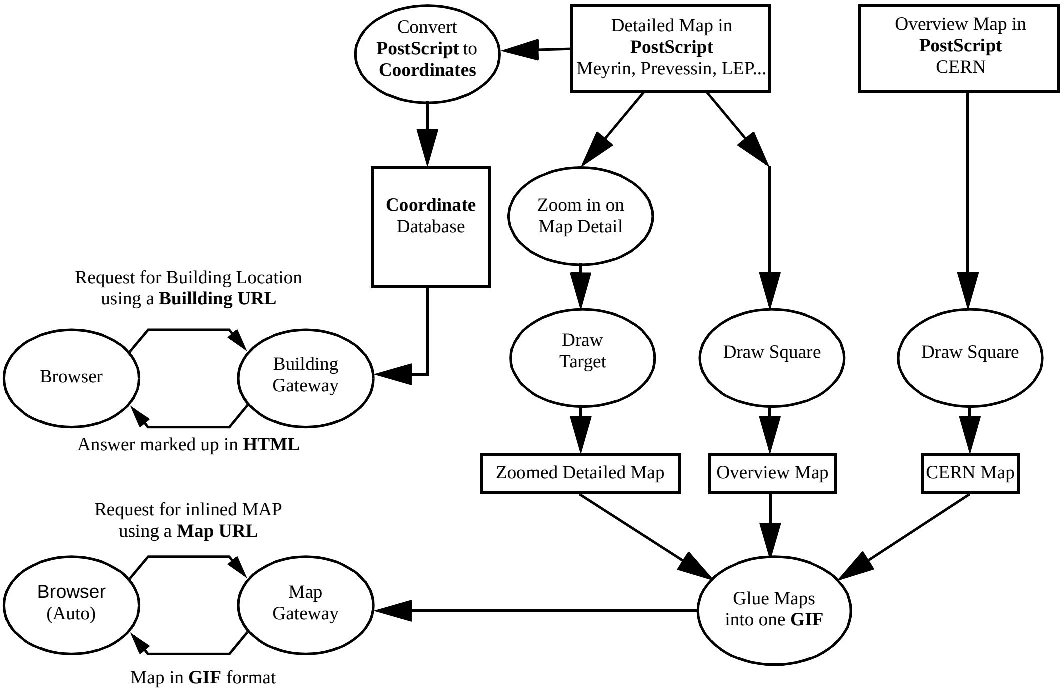

Convert **PostScript** to **Coordinates**
Detailed Map in **PostScript** Meyrin, Prevessin, LEP...
Overview Map in **PostScript** CERN
Coordinate Database
Zoom in on Map Detail
Request for Building Location using a **Buillding URL**
Browser
Building Gateway
Draw Target
Draw Square
Draw Square
Answer marked up in **HTML**
Zoomed Detailed Map
Overview Map
CERN Map
Request for inlined MAP using a **Map URL**
Browser (Auto)
Map Gateway
Glue Maps into one **GIF**
Map in **GIF** format

Figure 2: Dataflow of the web interface: the Building Gateway produces an HTML file from the lookup in the coordinate database, while the Map Gateway initiates the production of the three maps via GhostScript and glues them together using the NetPBM package.

3 The Palm Interface

The CERN maps are also available on a PDA, such as the Palm[2], see Figure 3. The Palm keeps all the PostScript maps, pre-converted, in memory and is able to draw, scale and translate them interactively. The drawing program as well as the converter are implemented in the Java language for portability reasons.

The Palm and other PDA's come with limited resources[9]. Though the Palm V has 2 Mbytes of memory, most of it is for database storage leaving only about 96 Kbytes for any program to run in. The Palm specific Java Virtual Machine, the kvm[10], uses up 32 Kbytes, the PalmOS uses around 16 Kbytes, which leaves any Java program with around 48 Kbytes of memory for objects and other storage. Screen size is limited to a 160x160 pixels with currently only 2 colors, but the latter restriction may be lifted soon with updates of the PalmOS system. The kvm, which is an implementation of the Java Micro Edition[11], implements the standard Java language with features such as multi-threading, but comes with a smaller API than any standard Java environment.

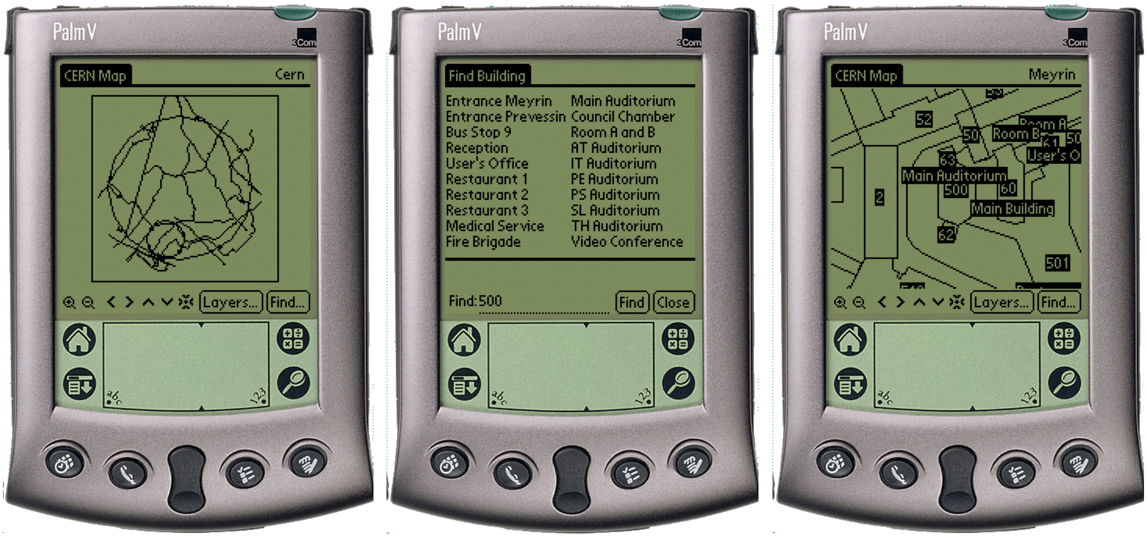


Figure 3: The Building and Map interface on a Palm V. On the left the initial screen showing the CERN site, clicking anywhere on the map will zoom in and show more detail. In the middle the lookup screen, allowing users to find common places and to look up building numbers, and on the right the result of a lookup of building 500, which can be scaled and panned interactively.

All PostScript maps run through a converter, using a standard Java environment on Windows or Unix, which builds a database for the Palm using the PDB format[12]. The converter compresses around 4.5 Mbytes of PostScript into a specific map PDB of 80 Kbytes. This database contains building names and numbers, coordinates of their location and all maps in the form of vectors, split up in several layers. The database format and record order is shown in Table 2,while the different record formats are shown in Tables 3-8.

Table 2: Format and order of the records in the map database

R#	RecordType	R/W[a]	Contents
0	Index	R/O	Version number and absolute indices to Map, Building and State records
1-...	State	R/W	Records the state of the application, scale factor, translation...
...	Drawable	R/O	Polygons for Map 0, Layer 0
...	Layer	R/W	Color, visibility and relative indices to Drawable records of Map 0
...	Drawable	R/O	Polygons for Map 1, Layer 0
...	Layer	R/W	Color, visibility and relative indices to Drawable records of Map 1
...	Building (sorted)	R/O	Building names, coordinates and relative indices to Map records
...	Map (unsorted)	R/O	Map names, bounding boxes and absolute indices to Layer records

a. R/W = Read/Write record, R/O = Read-Only record.

A GUI program of approximately 30 Kbytes provides different panels to display maps, to do lookups and to handle user interaction. It also handles the decoding of the database format. A binary search through the pre-sorted building names and numbers allows for fast access to any building's location. A conventional graphics engine[13] is used for drawing, implemented using long arithmetic rather than real arithmetic to enhance performance. The graphics engine draws vectors and text straight out of the database, thereby conserving memory usage. The resulting drawing can be scaled, translated, and the visibility of layers can be toggled interactively. Since a redraw may consume some time, the drawing engine runs in a separate thread to allow the user to interact instantly.

The development of the interface was done using a standard Java Development Kit (JDK) together with the kvm virtual machine and its libraries. The application was debugged and tested using POSE[14], a PalmOS Emulator running under Windows NT.

able 3: Format of the Index Record

Byte	Index Record
0	i (index)
1-2	length (19)
3-4	major version number
5-6	minor version number
7-8	map record start index[a]
9-10	map record end index[a]
11-12	building record start index[a]
13-14	building record end index[a]
15-16	state record start index[a]
17-18	state record end index[a]

[a] absolute reference

Table 4: Format of the State Record

Byte	State Record
0	s (state)
1-2	length (24)
3	state record type (1)
4-5	current map index[a]
6-9	current scale factor
10-13	current min. scale factor
14-17	current translation in x
18-21	current translation in y
22	current step for scaling
23	current step for translation

[a] relative reference

Table 5: Format of the Draw. Record

Byte	Polygon Drawable Record
0	d (drawable)
1-2	length (13+)
3-4	no of drawables
5	p (polygon)
6	width[a]
7-8	no of points for polygon 0
9-10	x coordinate
11-12	y coordinate
...	...
...	no of points for polygon 1
...	x coordinate
...	...

[a] currently only width=1 supported

Table 6: Format of the Layer Record

Byte	Layer Record
0	l (layer)
1-2	length (10+)
3	state (0=invisible, 1=visible)
4	color (0=white, 1=black)
5-6	start index of drawable record[a]
7-8	end index of drawable record[a]
9-...	layer name

[a] relative reference

Table 7: Format of the Build. Record

Byte	Building Record
0	b (building)
1-2	length (10+)
3-4	index to map record[a]
5-6	x coordinate
7-8	y coordinate
9-...	building name

[a] relative reference

Table 8: Format of the Map Record

Byte	Map Record
0	m (map)
1-2	length (16+)
3-4	start index of layer record[a]
5-6	end index of layer record[a]
7-8	bounding box x1 coordinate
9-10	bounding box y1 coordinate
11-12	bounding box x2 coordinate
13-14	bounding box y2 coordinate
15-...	map name

[a] absolute reference

4 Conclusion

Choosing Adobe ‘Illustrator’ PostScript as our single and only source of information for the CERN maps has proven to be very easy to maintain and keep up to date. After the initial Meyrin and Prevessin maps, other sites, such as the LEP and SPS pits, as well as new buildings have easily been added. Other larger sites or floor plans of libraries and computer centres could use the same interface for access from the Web or on the Palm.

The Web interface has been available for four years now and has proven its usefulness with an average of 50 lookups/day, peaking to over 200 lookups/day every year when the Summer Students arrive. The implementation in PERL using GhostScript was the right one at that time. Today an implementation in Java using Servlets may be faster and more portable.

The Java Palm interface was only recently made. The current pre-release of the virtual machine is not very stable yet and only runs on the Palm, though other PDA platforms may be supported in future. Todays Palms are still very low on resources, in memory as well as CPU. The interface is fairly slow at the moment, though a building lookup is faster than walking to a terminal and starting a web browser. Both the CERN Map application and the map database can be kept up to date easily with the “HotSynch” feature of the Palm. With more memory available the CERN phonebook could also be provided on the Palm, thereby making it easy to lookup peoples phone number and room location.

Acknowledgments

I would like to thank Robert Cailliau for encouraging me to create and deploy the Web interface to the CERN buildings as well as Herbert Mettler for re-drawing all the CERN maps in Adobe Illustrator and keeping them up-to-date. Thanks also to Péter Jurcsó, who recently updated the interface to cope with the whole set of LEP and SPS maps and who currently maintains the interface and Niels Dönszelmann for coining the term: chrono-chaotically.

References

1 CERN, “What’s the logic?”, Bulletin CERN, **50**/96.

2 Palm Computing, “Palm Connected Organizer”, 1999.
http://www.palm.com/

3 Adobe Systems, “PostScript Language: Tutorial and Cookbook”, Addison-Wesley, 1988.

4 Adobe Systems, “Adobe Illustrator 8.0”, Adobe Systems Incorporated, 1998.
http://www.adobe.com/products/illustrator/

5 Shishir Gundavaram, “CGI Programming on the World Wide Web”, O'Reilly & Associates, 1996.

6 Mark Dönszelmann, “Interfacing to the Web”, CERN Yellow Report, **95-05**, pp.295-306 (1995).
http://www.cern.ch/IPT/Papers/CSC95/Interfacing/

7 L. Peter Deutsch, “GhostScript, GhostView and GSView”, Aladdin Enterprises, 1999.
http://www.cs.wisc.edu/~ghost/index.html

8 Jef Poskanzer, “PBMPlus and NetPBM package”, 1994.
ftp://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/

9 Palm Computing Platform, “PalmOs Programmer's Companion (preliminary)”, 3Com, 1999.
http://www.palm.com/devzone/docs/palmos/

10 Sun Microsystems, “The K Virtual Machine (KVM)”, Sun Microsystems, 1999.
http://www.javasoft.com/products/kvm/wp/KVMwp.pdf

11 Antero Taivalsaari, Bill Bush, and Dough Simon, “The Spotless System: Implementing a Java System for the Palm Connected Organizer”, SMLI **TR-99-73**, Sun Microsystems, 1999.
http://www.sun.com/research/techrep/1999/abstract-73.html

12 Robert K. Fitterman, “The Pilot Record Database Format”, The Johns Hopkins University.
http://www.developer.com/directories/pages/roadcoders/pdb.html

13 James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, “Computer Graphics: Principles and Practice”, 2nd ed., Addison-Wesley, 1990.

14 Keith Rollin, “The Palm OS Emulator”, Handheld Systems **6.3**, May/June 1998.
http://www.cdpubs.com/hhsys/archives/63/12rollin.pdf