A WEB BASED AND TIME DEPENDENT VISUALISATION OF AN EVENT CALENDAR

M. HAMPE, K. HOSE*, C. HATGER AND C. KATTERFELD

University of Hannover, Institute of Cartography and Geoinformatics,
* University of Applied Sciences Oldenburg, Department Geoinformation

ABSTRACT

The University of Hannover maintains a service to request an event calendar for a certain time period. This paper describes an approach to communicate such temporal information adequately and intuitively through a dynamically generated webpage including an adaptive map. The geometries requested through a Web Feature Service (WFS) are transformed into an SVG image which allows for individual visualisations of the geometries. In our case, depending on the fact if and when an event is taking place, the objects of interest are highlighted in a certain manner. The buildings with an event today are more emphasised than the buildings with an event tomorrow.

In order to realise that, new ways for visualising spatial information have to be found. For example a multi-representation database (MRDB) serves as a basis for the geospatial data. This kind of database structure allows e.g. for multiscale maps, representing the object of interest with a higher resolution than the background objects.

The application serves for two main user groups: On the one hand for internet users working with a desktop PC, but on the other hand also a mobile device friendly version will be provided.

1 INTRODUCTION AND OVERVIEW

Today many kinds of information are ubiquitous to the user by use of desktop computers or mobile devices. Many of these information are geo-referenced. Thus, related information can be presented in a map or at least in combination with a map. An example is given by the web version of the “Yellow Pages”. Users can search for a certain business, e.g. restaurants or flower stores, at a certain place. The results of such a request are usually given by means of tabular listings. In addition a map is provided showing geocoded results by special symbolisation of locations matching the request. From a chronologically point of view neither request parameters nor subsequent results are capable of dealing with time depended information.

As web pages can be dynamically generated at the moment of the request, all necessary information can be collected from an up-to-date database. Hence, results can be visualised not only in a listing and a pre-generated static map but also in a dynamically generated map which takes into account all requested information. Regarding spatiotemporal information an example of such an application would be the presentation of spatial and chronologically encoded events. The University of Hannover offers such kind of information service, in terms of a calendar of all events taking place on the campus (University of Hannover, 2005). However, the current version of this event calender just offers a textual listing of all activities denoted by time, place and assigned faculty. Results are chronologically sorted by month. Hence the idea is both to combine spatial and temporal information within a single web based map. Additional information is arranged next to this, allowing for a more detailed explanation of the results displayed.

In this paper an extension to the textual presentation is given in terms of a dynamically created adaptive map. The user requirements for such a service are described in Section 2. Chapter 3 introduces the system architecture the service relies on. Section 4 describes the process of map creation from a cartographical point of view. The paper concludes with a summary and an outlook.

The system described here is based on results obtained from a diploma thesis that has been carried out with the Institute of Cartography and Geoinformatics during summer 2004 (Hose, 2004).

2 THE SYSTEM FROM THE USER’S POINT OF VIEW

Recall that we want to produce a depiction displaying spatiotemporal information that is based on a query. Therefore, the interface which is presented to the user should allow for some kind of interaction. To be more precise, at least those parameters which finally make up the result of a distinct query against a set of events and spatial data have to be provided by the user. In addition to this, results are to be displayed in a suitable fashion which enables the user quickly to derive the information she asked for. For the sake of simplicity, this approach combines both information related to user input and system output into one single web page.

2.1 Required user input

Any event which is published by the calendar is affiliated to the university. As a consequence the major part of users is expected to be related to the university, as well. A typical user would either be student, lecturer or research associate. They all own different technical attitudes. Thus, a potential user should be enabled for constrained search requests limited to his own field of interest. The University of Hannover is organised by means of faculties. It is obvious to use this organisational structure of faculties in order to express some special field of interest. If there is no idea about which faculty to choose for some event, users are allowed to perform full search over all organisational units available to the system so far.

Additional required parameters are given by means of temporal propositions. Any search for some event should be constrained by some period limiting search space. In this case users are engaged to provide values denoting begin and end of search space. In practice this is done by entering year, month, day, hour and minutes into the appropriate input boxes. If nothing at all is entered default values using the current date are used.

Furthermore the results are to be displayed in an adaptive map. “Adaptive” means in this context that the resulting map depends on the user’s choice of events as well as the choice of the output format. The user has the choice between different styles of the map, dependent on the device she is using, e.g. a desktop PC, mobile device like PDA or Smartphone, or a printable map.

Finally, the map display can be controlled by the user through selection and deselection of map layers containing different thematic information to adapt the density of information needed. If no choice is made any available layer is returned to the user.

Concluding all the user interface requires the following functionalities:

The user is allowed to select the

- time span of interest,
- field of interest,
- map layers and
- output format.

The resulting application should offer the following content:

- Campus map with objects hosting an event highlighted
- Legend of the map
- Additional information about the events
- Help functionalities

The resulting user interface can been seen in Figure 4. For a detailed description see Section 3.3.

3 SYSTEM ARCHITECTURE

The system relies on a multi-tiered architecture. The first tier being the client by means of a web browser. Suitable browsers must be able to interpret JavaScript and SVG (Scalable Vector Graphic, SVG 2001) as will be shown later on. Any request submitted by a client is sent to the second tier which extracts parameters provided with this query and performs input validation. In addition, the second tier behaves like a mediator. It forwards subsets of request parameters to subsequent tiers as well as it combines resulting information returned from those. Evaluation of temporal and field of interest request parameters is carried out by the third tier. Thus, it returns a set of events matching the request parameters to the second tier. The result set carries additional attributes like associated spatial object identifiers (SOID). Selected output format and map layers as well as the SOID mentioned before are registered with the fourth tier which then performs cartographic map production of spatiotemporal events. Any server side tier is realised as a servlet that complies to the J2EE standard specifications (SUN 2005). Therefore, this approach makes use of a suitable application server.

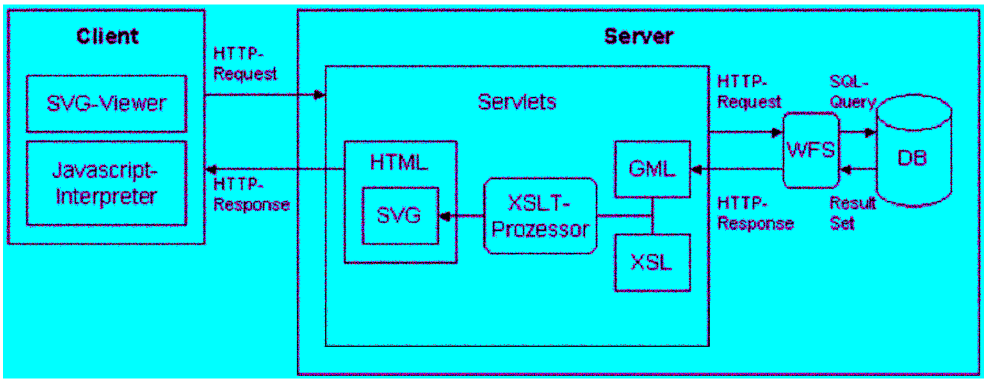


Figure 1. Client/Server Architecture of the service (Hose 2004).

Figure 1 shows both data flow and system architecture. Communication between each tier is realised via utilisation of (stateless) hypertext transfer protocol (http). Session management is used to make request specific information available to different components. Data returned to the client comprises both map and additional illustrated information. The process of map generation is shown in more detail in Section 3.2.2. Spatial data is requested from a web feature service (WFS) which sub sequentially retrieves raw data from a relational database management system (RDBMS). Raw data is transformed by WFS into geography mark-up language (GML) – some specialisation of the extensible mark-up language (XML) – and returned to that servlet which performs map production. Accordingly, GML is converted into scalable vector graphics format (SVG) and delivered to the client finally. This is done by applying an Extensible Stylesheet Language (XSL) transformation to the GML data. In the following section the major parts of the system are described in more detail.

3.1 Data model

Two different types of data are used in the application: the event data, which is primarily of temporal nature and spatial, non temporal information. In this approach an event is characterised by several distinct properties. These are stored in a RDBMS. Figure 2 shows a class diagram of the conceptual data model. It uses the unified modelling language (UML) notation.

In detail the relation concerning the events holds the following attributes:

- event identifier,
- name,
- type,
- description,
- beginning of appointment,
- end of appointment and
- designated SOID.

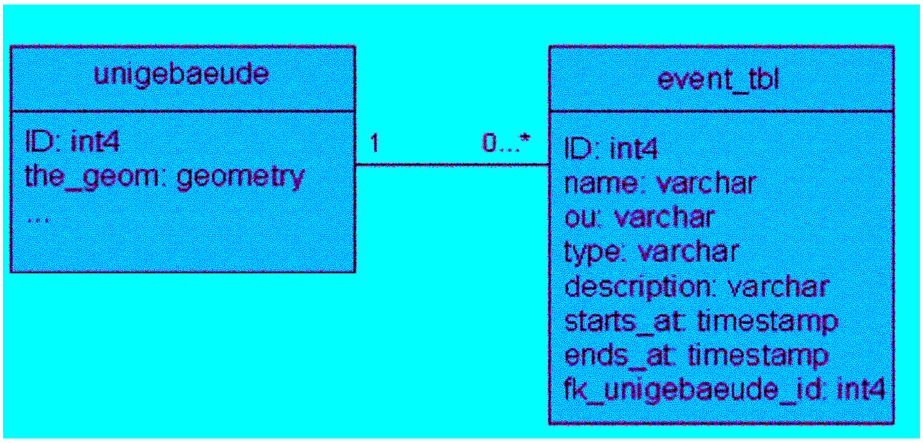


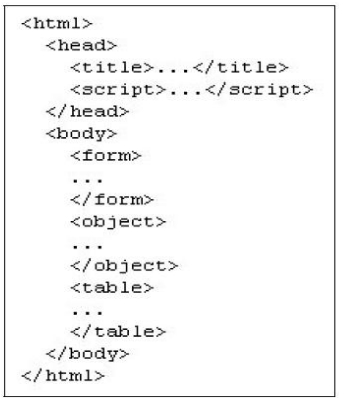
Figure 2. UML class diagram showing conceptual data model (Hose 2004).

Currently the system owns a simple n:1 mapping of events to SOIDS. This is events can only take place at single SOID. In principle this is not feasible at all. A more sophisticated solution would allow for n:m mappings of events to locations.

Geodata that have been used is stored in a RDBMS which can handle spatial information. In this approach we rely on the open source RDBMS PostgreSQL (2005). PostgreSQL provides native support for geometries like points, lines and areas. In addition to this the open source extension PostGIS (2005) has been used. This allows for management of georeferenced data within the RDBMS. The geodata used here are topographic data (ATKIS-Base-DLM). These data are provided by the National Mapping Agency of Lower Saxony. As ATKIS (2005) does not contain any information on buildings, these haven been taken from a cadastral dataset. A unique identifier is assigned to each building. Thus this allows for interrelating a certain building to one or more events.

3.2 The server environment

The coarse workflow generating the web page dynamically has been described above. As a result of the request the user receives an HTML-document. The visual output of such document can be seen in Figure 4. The HTML-document consists of several parts, whereas each part of the document is generated by a corresponding servlet. Therefore the HTML–code is generated piecewise, starting with the header of the document, the form-tags, the SVG-image, the map-legend and the event table (cf. Figure 3).



```
<html>
  <head>
    <title>...</title>
    <script>...</script>
  </head>
  <body>
    <form>
    ...
    </form>
    <object>
    ...
    </object>
    <table>
    ...
    </table>
  </body>
</html>
```

Figure 3. The HTML-document structure including header, input forms, SVG-object and resulting table (Hose 2004).

By requesting the event calendar the user accesses the main servlet which organises the orchestration of the other classes and servlets, which in turn are responsible for a certain part of the HTML-document the user receives as a result (in Figure 4 the whole map can be seen; the generation of the individual components is described in the following).

3.2.1 User input form

After evoking the service by sending a request including the parameters listed above, the servlet container forwards the request to the *EventMapServlet*, which on his part forwards the request to the servlets generating the parts of the HTML-code. The *BannerServlet* generates the header, including the JavaScript functionalities, followed by the *QueryFormHeader*, which completes the header part.

The next step is to generate the input form for requesting the time span, area of interest, map layers and output style, which is handled by the four servlets *FrmPartAppointment*, *FrmPartOranizationalUnit*, *FrmPartStyledThemes* and *FrmPartStyles*.

The input form has to be generated based on a configuration-file (*config.xml*) and at the same time it has to reflect the user's input data. That means the servlets have to determine the configuration of the web-application and at the same time pass on the input data (*EventMapProperties*).

Without going into too much details of the event and context handling of the servlets, the following can be stated: Generally, the workflow for generating the input form is to determine and process the data, sent along with the request, store these data as attributes in the session-object *EventMapProperties* and generate the HTML-code regarding the configuration.

The *FrmPartAppointment*-Servlet, which generates the input form of the time span, first determines the parameter for time and date sent with the request, checks the validity of the data and forwards them to the

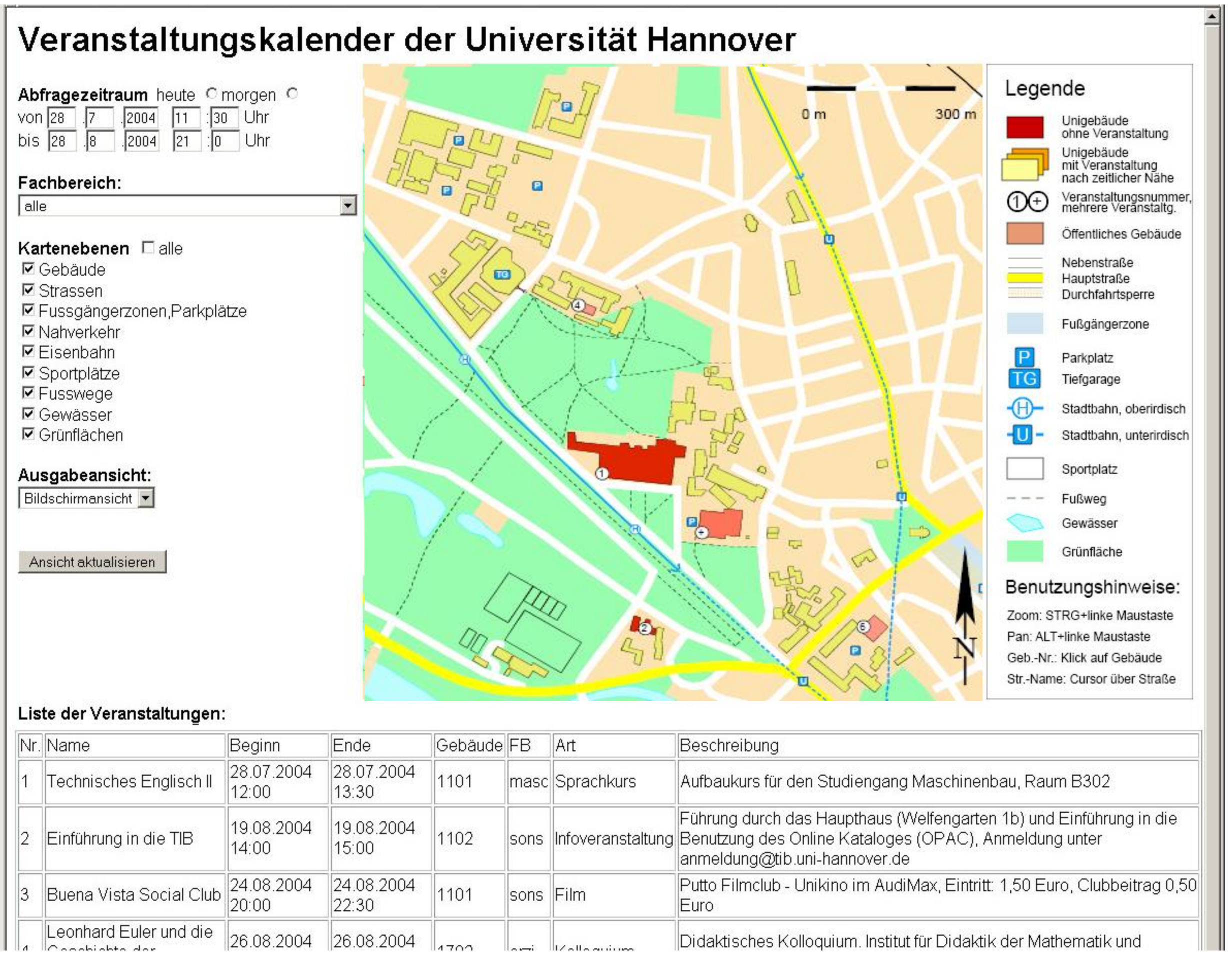
EventMapProperties-Object. After that the HTML-code for the input form is generated and the actual values for the date are inserted. Additionally the user has the possibility to select “today” or “tomorrow” using RadioButtons. The resulting input form can be seen on the top left of Figure 4.

FrmPartOrganizationalUnit generates a list for selecting the departments of the university. It requests the *EventTblConfig*-Object of the *ServletContext*, which maintains the available faculties and forwards the selected entry to the Session-Attribute *EventMapProperties*.

FrmPartStyledThemes generates Check Boxes of the available map layers. Again the *EMapConfig*-File is requested for displaying the layers regarding to the *LayerMapping*-Object, enabled during the configuration process. The parameters sent by the user are stored again in the *EventMapProperties*-Object. During the first call only building and roads are selected. That is because during the first call it should not take too much time loading the default SVG-map.

Similarly, the different ways of presentation are generated in the *FrmPartStyles*-Servlet. The options are colour screen or black-and-white map, optimised for printing. After determining the user’s selection and storing these attribute in the *EventMapProperties* the list will be generated according to the configuration stored in the *EMapConfig*-Object. As default “colour screen” is selected.

After generating the parts responsible for the data input, the *QueryFormTrailer*-Servlet adds the “Update” button for sending the form and after that the form will be closed by its end tag.


Veranstaltungskalender der Universität Hannover

Abfragezeitraum heute ○ morgen ○
von 28 . 7 . 2004 11 : 30 Uhr
bis 28 . 8 . 2004 21 : 0 Uhr

Fachbereich:
alle

Kartenebenen □ alle
☑ Gebäude
☑ Strassen
☑ Fussgängerzonen,Parkplätze
☑ Nahverkehr
☑ Eisenbahn
☑ Sportplätze
☑ Fusswege
☑ Gewässer
☑ Grünflächen

Ausgabeansicht:
Bildschirmansicht

Benutzungshinweise:
Zoom: STRG+linke Maustaste
Pan: ALT+linke Maustaste
Geb.-Nr.: Klick auf Gebäude
Str.-Name: Cursor über Straße

Liste der Veranstaltungen:

Nr.	Name	Beginn	Ende	Gebäude	FB	Art	Beschreibung
1	Technisches Englisch II	28.07.2004 12:00	28.07.2004 13:30	1101	masc	Sprachkurs	Aufbaukurs für den Studiengang Maschinenbau, Raum B302
2	Einführung in die TIB	19.08.2004 14:00	19.08.2004 15:00	1102	sons	Infoveranstaltung	Führung durch das Haupthaus (Welfengarten 1b) und Einführung in die Benutzung des Online Kataloges (OPAC), Anmeldung unter anmeldung@tib.uni-hannover.de
3	Buena Vista Social Club	24.08.2004 20:00	24.08.2004 22:30	1101	sons	Film	Putto Filmclub - Unikino im AudiMax, Eintritt: 1,50 Euro, Clubbeitrag 0,50 Euro
4	Leonhard Euler und die Geschichte der	26.08.2004	26.08.2004	1702	erzi	Kolloquium	Didaktisches Kolloquium. Institut für Didaktik der Mathematik und

Figure 4. Resulting web page including the input form (left), SVG map (right) and the events listed in a table (bottom).

3.2.2 Generating the map

The second part of the HTML-page, besides the input form, is the resulting map, an SVG image. The *SVGTagServlet* gets the HTTP-request and takes care of generating and embedding the SVG-image into the HTML-page. The attributes of the actual session are stored in the *EventMapProperties* and can be read out by the *SVGTagServlet*.
The next step is to collect the event data from the database. To this end the *ServletContext* is requested, which contains the object *UnikarteDBAO* for generating a database connection via a JDBC (Java Database Connectivity)-interface. This object includes a method for generating and sending an SQL-request to the database, depending on the time span and area of interest, set by the user, resulting in a list of events and their attributes, matching the request.
With the configuration (*EMapConfig*), accessible through the *ServletContext*, the user input data (*EventMapProperties*) and the event data (*EventDetails*) coming from the database, all the necessary information have been collected for generating the SVG map. The *SVGProducer* generates an HTTP-request, depending on these data, sent to the WFS-implementation. The WFS-request consists of the service address (taken from *EMapConfig)* and the map layers. Based on these information the WFS requests the geodata from the database and returns a GML-file including the requested geodata.
Depending on the presentation style selected by the user, a stylesheet for transforming the GML data into an SVG image is selected. The information about the XSL file to select are stored in the *EMapConfig* element.
Before transforming the data into a map image, an appropriate visualisation of the events is set up (see Section 0). After the transformation the SVG-code is returned to the *SVGTagServlet* and is integrated into the HTML-page as an <object> element, together with the map legend.

Finally a table listing all relevant events is integrated into the HTML-document. This table is generated by the *ListEventsServlet*. For this purpose again the *UnikarteDBAO*-Object is requested, which accesses the data from the database. The request properties used in the database call accord to the attributes stored in the *EventMapProperties* object, which reflects the user input data. The events are listed in the table chronologically (cf. Figure 4, bottom).

4 Visualisation of Map Data

The advantage of the event calendar map is its dynamic generation process. As the map is generated at the moment it is requested all the actual information as well as the requirements of the user can be taken into account. The map coming along with the event calendar is individualised depending on the user's selection. She can determine the map layers, the map style as well as the content and date of the events of interest.
The challenge is to communicate all the information about the real world environment as well as the location and time of an event at a glance. The approach to fulfil these requirements will be presented in the following section.

The layout of the whole map is dependent on the output device. In our methods we take into account a desktop PC, a mobile device, i.e. a PDA, and a printed black-and-white map. Depending on the user's selection, different transformation rules for generating the SVG output map are used. In the case of the multiscale map (cf. Section 4.2) additional servlets are involved for processing the geodata.
The main information of the event map are the place and the time of the event the user has chosen. The possible locations are reflected by the university buildings, stored as polygons in the database. To localise the objects hosting an event, these buildings have to differ from the objects with no event. At the same time these map objects have to catch the map reader's attention.
To separate the map objects from each other and to focus the map reader's attention the cartographer uses the graphic variables named by Bertin (1983). Bertin suggests that there are seven fundamental visual variables. The variables include the geographic position of a mark on a surface, which can vary in two dimensions (x and y location). A particular mark can further vary graphically in size, value, grain, colour, orientation and shape. Nowadays also transparency or frequency of an animated object can be used as a graphical variable in a digital map.

4.1 Using graphical variables for emphasizing buildings of interest

The visual presentation of the spatial data is realised by the SVG vector format. The SVG specification is an XML technology, developed by the World Wide Web Consortium (W3C) with the purpose of creating a standardised vector graphics format for the web environment. The SVG format is widely seen as the obvious future standard for all kinds of vector graphics on the web. Currently SVG images can be visualised in a web browser, e.g. by using an Adobe provided SVG Viewer plugin (Lehto et al., 2003).

Beside the standard visualisation variables the SVG technology in combination with JavaScript provides a set of additional possibilities like animation or mouse over effects. These effects help the user to localise the events, i.e. answering the “where”.
Beside the “where” we want to answer the question of “when” in the map. The user should be able to determine the relative temporal distance of the event. To express the different quantities, the variation of the following variables would be appropriate:

- Hue (the closer the darker)
- Transparency (the closer the less transparent)
- Grain
- Frequency of animation

Also a combination of two variables, e.g. an animation only for the closest event, the others visualised by a variable hue, is imaginable. Additionally font is used to express the absolute date, used in an mouse over effect.

At the same time two aspects have to be visualised in one map with the help of these graphic variables: To separate the buildings hosting an event from all the other buildings and within these selected objects to determine the events closer in time than the others. A combination of two variables is necessary to communicate this information. To express an event taking place in our example map we used different colours, yellow for an ordinary building, red for the ones hosting an event. For the distance in time a variation in hue, transparency or alternatively frequency was chosen.
But the length of the graphic variables is limited. For instance the number of the variation of hue should be not more than eight in a colour map (Bertin, 1983). Otherwise the user is not able to associate a certain value of hue with a certain time. That means the number of classes can’t be the same as the number of events, as the number of events is variable and could be more than eight. Because of that before visualising the resulting events they have to be combined into classes. Again, there are different possibilities for combining a set of numbers into classes, as the length of the class could be constant (e.g. one day), or the number of elements per class, or the number of classes could be constant, as in our case. We decided to build only three commensurate classes.
As the time span is variable, depending on the user’s input, a method was implemented to separate the resulting events into these equally sized classes, whereas a static separation has the disadvantage of cutting at a inauspicious time. E.g. a strict separation would result in a class starting at 14:00 and ending at 23:20, as the user would expect a time span ending at 0:00 o’clock.

The different representations of the buildings are realised through the transformation from GML to SVG. The XSL document contains the information how to translate an element of the GML file into an element of the SVG file. Within the XSLT (Extensible Stylesheet Language Transformation) file, parameters are defined that contain a list of the IDs of the buildings hosting an event, separated by the time class.
Additionally, the event numbers, according to the numbers in the event table, are shown (cf. Figure 4). This can be realised as the IDs of buildings having an event are stored in a list chronologically, as mentioned above. When clicking on the symbol the user gets all information available for this particular event. If more than one event is taking place this information is expressed by a “+” instead of the number of the event.

4.2 Using multiscale information

Beside the methods of emphasising map objects mentioned above, an alternative approach is to highlight the objects of interest using a representation with a higher level of detail (LoD), more detailed than the objects of minor interest. This approach had been used in earlier applications to balance the disadvantage of small displays of mobile devices and the required information content (Sester et al. 2004). As the space on these small displays is limited also the capacity of information which can be shown is restricted. That leads to the idea of showing more information of the objects of interest, like in our case the buildings hosting an event, and at the same time presenting the other objects with a lower LoD. Information is filtered, because otherwise the map would be overloaded with too many information. To filter information in cartography, generalisation is used. Information can be filtered by leaving out some objects completely, or just simplifying the objects of less importance.
This generalisation can be done at the moment the map is requested, which works for just filtering objects or classes of objects, but as generalisation is a time consuming procedure, a second approach which falls back on pre-generalised data is more efficient.
A multiple representation database (MRDB) serves as a basis for the detailed as well as the less detailed information. In the following section the concept of an MRDB will be shortly described.

4.2.1 The MRDB approach

A multi representation database (MRDB) can be described as a spatial database, which can be used to store the same realworld-phenomena at different levels of precision, accuracy and resolution (Devogele et al., 1996; Hampe et al., 2003). Besides this multi-scale aspect, it can also be understood in terms of multiple representation by focussing on information of different thematic disciplines.

There are two main features that characterise an MRDB:

- Different levels of detail (LoD) are stored in one database.
- The objects in the different levels are linked.

The first feature can be compared to the analogue map series of the National Mapping Agencies: These maps of different scales exist separately, only implicitly linked by the common geometry. In the second case, however, individual objects are explicitly linked with each other and thus each object "knows" its corresponding objects in the other representations.

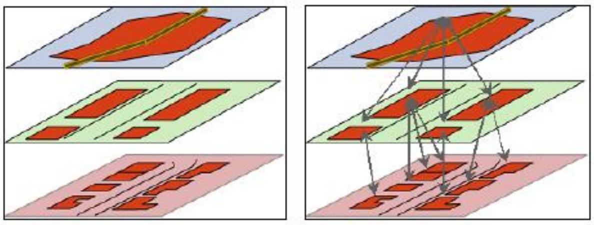


Figure 5. The MRDB principle: Several representations are stored in a database (left), corresponding objects are linked (right).

The user is interested in certain objects, in our case the buildings having an event, more than in other objects. To emphasise these objects in the map several possibilities are cogitable. In order to visualise one individual building in the generalised data set a possibility is to use graphic variables like colour (Figure 6 left), size or the shape of the object. If, however, in a certain scale the objects are generalised, the individual objects can no longer be distinguished. Buildings could e.g. be amalgamated which in turn means that we cannot localise a single building to point out an object of interest. Instead the whole building block is highlighted (cf. again Figure 6 left). Alternatively, the position of the objects of interest can be marked with a dot or special symbol (Figure 6 right). This however hides underlying structures and prevents an exact identification of buildings. Furthermore the target object still cannot be localised exactly.

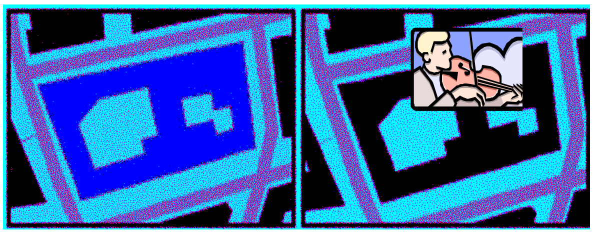


Figure 6. Emphasising objects of interest, using: colour (left), symbols (right).

Thus, a combination of this individual building and the generalised background buildings is desirable, given in Figure 7 (left) with a detailed representation of the target objects and at the same time aggregated and simplified representations of the other buildings. In this way, different LoDs are given in one representation.

A similar approach is used by Sester (2002), where certain objects are emphasised by a discrete change of scale: Objects can be enlarged (cf. Figure 7 right), kept at high detail, whereas background objects are simplified or even aggregated.

The approaches mentioned above are mainly interesting for maps dedicated to small displays like the ones of PDAs or Smartphones. In the case of our event calendar, the buildings would be represented in a simplified manner and only those hosting an event would be highlighted using a higher level of detail (Figure 7 left).

The simplified presentation of the environment would keep a clear map view, also on small displays, and on the other hand the user gets all the necessary detailed information of the objects of interest, which helps for orientating and recognising the object of interest in the reality. Practically, the production of the map is the same as described in Section 3.3.2. The only difference is that the objects are now requested from the MRDB (instead of the database), and additionally every requested object knows its desired level of detail. The MRDB system, in turn, extracts all the relevant data from the database. As the different representations of the same real world object are linked with each other, the user or the application can choose the desired representation of objects. That means the higher the resolution of the display the more detailed representation can be used in the map. More details about setting up and using an MRDB system can be found in Hampe et al.(2004).

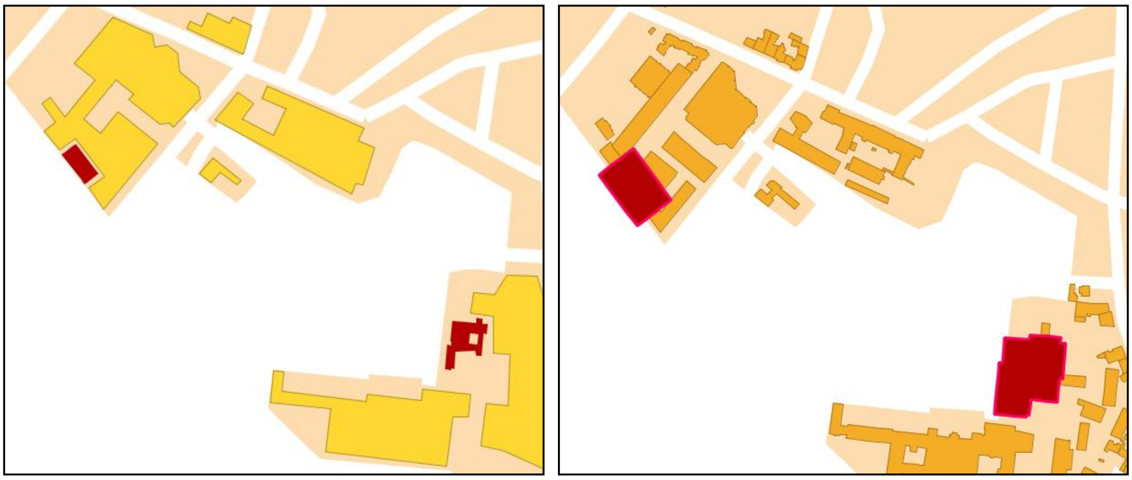


Figure 7. Emphasising objects hosting an event by using different LoD (left) or amplifying the objects of interest (right).

5 SUMMARY & OUTLOOK

Events of an organisation or a city are typically given in terms of lists in paper form or in the internet. Mostly, these lists are sorted in chronological order. All events are inherently linked to places. In conventional calendars, this is given by the addresses. The idea of the event calendar presented in this paper has a twofold advantage as opposed to the traditional ones: it offers at a glance both the location of an event and the closeness in time. This is an elegant and efficient way of communicating temporal information that has important spatial aspects. Such options are possible only with today's technology. The example in this paper has shown the power and possibilities of today's map making as the map is generated dynamically, falling back on the actual information stored in the database and furthermore taking the special request of the user into account.

We presented some traditional ways of visualisation as well as new possibilities to emphasise certain objects in a map. Ideas of using an MRDB not just for storing and organising spatial data, but also for presenting the geodata in a map using different representations of the map objects. The latter also in respect to the small displays of mobile devices. Regarding the visualisation the SVG technology provides also new possibilities, more than simple rastermaps. The visualisation aspects have not been exhausted. More possibilities of visualising the events were possible, but this work gave a first impression of the possibilities and also showed a way to generate a personalised map dynamically.
Beside the visualisation aspects we introduced the technical architecture of generating a dynamic web page including an SVG map, accessible with the help of a desktop computer used at home as well as a mobile device.

The future work will concentrate on the aspects of a service for mobile devices including the small display cartography. A few ideas of generating maps dedicated to the small displays have been illustrated in this paper. In the near future a client application will be set up for visualising SVG maps on mobile devices. This application will be used to request the different representations of the event map to find out which representation meets the user requirements and the small display of the mobile device best.
Furthermore the implementation presented in this paper, which is currently just a prototype running offline, will be integrated into the web page of the University of Hannover in the near future.

REFERENCES

ATKIS, 2005. Authoritative Topographic Cartographic Information System. At: http://www.atkis.de (accessed April 30th, 2005).

Bertin, J., 1983. Semiology of Graphics, W. J. Berg, Transl. The University of Wisconsin Press, Madison Wisconsin.

Devogele, T., Trevisan, J. and L. Raynal, 1996. Building a Multi-Scale Database with Scale-Transition Relationships. In: M. J. Kraak and M. Molenaar (Eds.), *7th International Symposium on Spatial Data Handling*, Delft, The Netherlands, Taylor & Francis, pp. 6.19-6.33.

GML, 2003. Geography Markup Language Implementation Specification, Version 3.0, at https://portal.opengeospatial.org/files/?artifact_id=7174 (accessed April 30, 2005).

Harrie, L., Sarjakoski, L. T. and L. Lehto, 2002. Variable-Scale Map For Small-display Cartography, *Proceedings of ISPRS*, Vol. XXXIV, Part 4, Ottawa, Canada, pp. 237-242.

Hampe, M., Anders, K.-H. and M. Sester, 2003. MRDB Applications for Data Revision and Real-Time Generalisation, *Proceedings of 21st International Cartographic Conference,* 10. - 16. August 2003, Durban/South Africa.

Hampe, M., Sester, M. and L. Harrie, 2004. Multiple Representation Databases to Support Visualization on Mobile Devices. Proceedings of the XXth ISPRS Congress, July12-23, 2004, Istanbul, Turkey, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXV(B4:IV), pp. 135-140.

Hose, K. 2004. Webbasierter Veranstaltungskalender mit Raum- und Zeitbezug (Web based event calendar with reference to place and time), Diploma thesis. An electronic version available at http://www.ikg.uni-hannover.de/publikationen/diplomarbeiten/2004/hose/Diplomarbeit.pdf (accessed April 15, 2005).

Lehto, L., 2003. Architecture specification. GiMoDig-project, IST-2000-30090, Deliverable D4.4.1: Final system architecture, Public EC report, 41 p.

PostGIS, 2005. PostGIS, Geographic Objects for PostgreSQL. At http://postgis.refractions.net (accessed April 27, 2005).

PostgreSQL, 2005. PostgreSQL. At http://www.postgresql.org (accessed April 27, 2005).

Sester, M., 2002. Application Dependent Generalisation - the Case of Pedestrian Navigation, *Proceedings of the Joint International Symposium on "GeoSpatial Theory, Processing and Applications", ISPRS/Commission IV, SDH2002,* Ottawa, Canada, July 8-12, 2002, 6 p, CD-ROM.

Sester, M., Sarjakoski L. T., Harrie, L., Hampe, M., Koivula, T., Sarjakoski, T., Lehto, L., Elias, B., Nivala, A.-M. and H. Stigmar, 2004. Real-time generalisation and multiple representation in the GiMoDig mobile service. GiMoDig-project, IST-2000-30090, Deliverables D7.1.1, D7.2.1 and D7.3.1, Public EC report, 151 p. An electronic version available at http://gimodig.fgi.fi/deliverables.php (accessed April 15).

SUN, 2005, Sun Microsystems Java 2 Platform – Enterprise Edition (J2EE), at http://java.sun.com/j2ee/index.jsp (accessed April 30, 2005).

SVG, 2001. Scalable Vector Graphics 1.0 Specification, at http://www.w3.org/TR/SVG (accessed April 30, 2005).

University of Hannover, 2005. The event calendar of the University of Hannover at http://www.uni-hannover.de/aktuell/veranstalt.htm (accessed April, 25, 2005).

WFS, 2002. Web Feature Server Implementation Specification, Version 1.0.0, at https://portal.opengeospatial.org/files (accessed April 30, 2005).

BIOGRAPHY

Dipl. Ing. Mark Hampe, born in 1973, studied Geodesy at the University of Hannover and obtained the Master’s degree (Dipl.-Ing.) in 1999. His master thesis was about „animated maps and their advantages for the user”. Between 1999 and 2001 he attended a trainee on public service. Since 2002 he is a scientific assistant at the institute for cartography and geoinformatics, University of Hannover. His primary exercise lies in the research fields of MRDB as well as visualisation of spatial data on mobile devices.