A Federation Architecture for an Environmental Information System incorporating GIS, the World Wide Web, and CORBA

Arne Koschel, Ralf Kramer, Ralf Nikolai
Forschungszentrum Informatik (FZI)
Haid–und–Neu–Str. 10–14, D–76131 Karlsruhe, Germany
{koschel, kramer, nikolai}@fzi.de

Wilhelm Hagg, Joachim Wiesel, Heiko Jacobs
Institut für Photogrammetrie und Fernerkundung (IPF)
Universität Karlsruhe, D–76128 Karlsruhe, Germany
{hagg, wiesel, jacobs}@ipf.bau-verm.uni-karlsruhe.de

Abstract

In environmental information systems two important problems have to be solved. First, environmental questions touch upon a large number of fairly different knowledge domains. These questions are normally addressed by different systems, which are usually heterogeneous, both technical and semantical. Therefore, in order to answer environmental questions, e.g., in state authorities, data and functionality from different sources have to be available uniformly. Second, environmental information kept by authorities should be available for the public as well.

Nowadays, emerging standards for open, distributed systems like the Object Management Group's Common Object Request Broker Architecture cope with the first problem. Especially on a global scale, the World Wide Web (WWW) and its tools tackle the second problem.

For these purposes, a WWW-based application has been developed that enables clients without specific software tools to get geographic information and business graphics for visualizing tabular data. Maps and graphics can be combined to create thematic maps or to down-load maps and graphics for further processing at the client site. In a first implementation, we use the GIS GRASS and GNUPLOT with some extensions as services to provide map and business graphics pictures. In this paper we present the implementation of our GIS access services and we show how this fits nicely into our federation architecture.

1 Introduction

In large environmental information systems (EIS) two important problems have to be solved. First, environmental questions touch upon a large number of fairly different knowledge domains. These questions are normally addressed by different autonomous systems, which are usually quite heterogeneous, both technical (e.g., different operating systems, programming languages etc.) and semantical (e.g., different data models). Therefore, in order to answer environmental questions, e.g., in state authorities, data and functionality from different sources have to be available uniformly. Second, environmental information kept by authorities should be available for the public as well.

Nowadays, emerging standards for open, distributed systems like the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) cope with the first problem. Especially on a global scale, the World Wide Web (WWW) and its tools tackle the second problem.

A focal point in EIS are Geographic Information Systems (GIS), which provide geographic data and maps for environmental modelling and statistical data analysis. Integrating GIS objects is a difficult task, since there are several standards for data formats and a wide field of methods for data analysis. A standardization of data formats is currently under way within the OGIS (Open Geodata Interoperation Specification) effort. However, no standard is available yet.

The remainder of this paper is organized as follows: In Section 2 we describe the federation architecture, which is based on the World Wide Web and CORBA. Section 3 describes in some more detail how GIS functionality and visualization facilities are implemented. Section 4 elaborates on using CORBA for accessing relational databases, which are among the most important information sources in EIS. A survey on related work is given in Section 5. Section 6 presents the conclusions and gives an outlook on future work.

2 Outline of the Federation Architecture

Figure 1 depicts a high-level view of the architecture. We assume that the reader is familiar with the World Wide Web (WWW). A short introduction to WWW and the Common Gateway Interface is given in [McCauley 96]

Horizontally, we distinguish between application level and system level services. System level services provide basic functionality such as accessing databases and preparing HTML pages. Application level services basically combine several system level services into higher level services that are available for the user. Using CORBA, these services can be transparently assigned to computers in a network.

Vertically, we distinguish between information and data services. The information services help the user to find relevant data sources, e.g., data services and reports. They are based on the environmental data catalogue UDK. This metainformation system is used in most German states and in Austria. For an in-depth description of the current development of the UDK we refer to [Lessing 96]. Our first prototype made UDK data available using the tools and techniques of the World Wide Web [Kramer 95, Kramer 96]. Meanwhile, we extended the underlying UDK data model in order to provide online access to data services as shown in figure 1. These data sources primarily are environmental databases and reports that are as well accessible with WWW tools. Data sources may comprise geographic information systems (GIS), relational, pre-relational, object-oriented database systems, and expert systems. Data services deliver results that in turn are used as input for the visualization facilities described in Section 3.

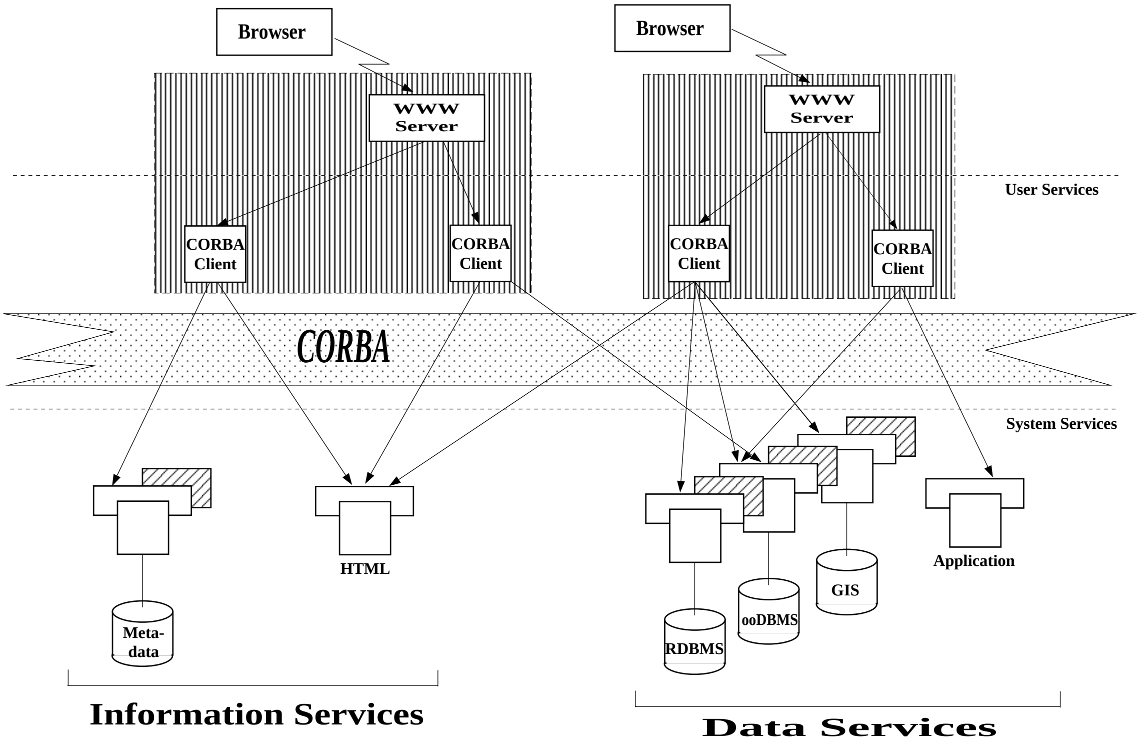

Figure 1: Overall Software Architecture

3 Visualization Facilities

3.1 Requirements

Visualization of datasets is an essential task in all information systems. Requirements to visualization tools increase with the complexity of the data. In EIS, it is absolutely mandatory to handle both tabular data and complex geometric data; tabular data have georeferences as well. Both data types have to be visualized such that both an overview of all data and – on user's request – details of the data are presented.

Several GIS provide such facilities. An overview of existing GIS can be found in [Rodcay (ed.) 95]. However, these GIS are fairly complex to handle. Hence, they are inappropriate for the casual user. Furthermore, these GIS consume huge amount of resources (e.g., main memory) and are pretty expensive.

3.2 General Approach and Tool Selection

In the architecture of figure 1, we use a GIS and other visualization tools at the WWW-server site to meet those requirements. At the client site, only a standard WWW browser is required. All tools used are in the public domain; hence, there are no licence fees incured at the server site. In the remainder of this Section, we describe those tools and the techniques to connect them to WWW.

The implementation of our services is based on Tcl [Ousterhout 94], a high level command interpreter that provides easy access to files and convenient string manipulation. Tcl as well offers many different data types such as lists and associative arrays.

In order to extract maps from the GIS and to provide complex GIS functionality, we use the public domain tool GRASS, which is a raster oriented GIS that supports digital image processing, map creation and several vector operations. GRASS is implemented in C and is available on several UNIX platforms. Using a public domain GIS enables source code level debugging and the extension of the existing pool of methods, since the source code is available [U.S. Army Corps of Engineers 95]. In addition, the GRASS mailing list is a good forum for the discussion of problems [GRASS 95].

The visualization of tabular data requires a tool for generating business graphics. Currently, we use GNUPLOT [GNU 95] to create line-, bar-, and pie-charts. Since GNUPLOT is originally a program designed for scientists to plot data and mathematical functions, it does not match the requirements exactly. To draw bar charts, e.g., is not very effective. Extensions to GNUPLOT are restricted to a given version of GNUPLOT; there is no guaranty for a support in the future. Other programs, which might match the requirement better, are to our knowledge currently not available in the public domain.

In addition, we need a tool to combine outputs from different raster data sources such as maps, diagrams and icons, and for different format conversions between standard raster image formats like GIF, PPM and TIFF. We use Tom Boutell's gd-library [Boutell 95] for operations on GIF-images. The FLY-tool [Gleeson 95] from Martin Gleeson, which is a commandline interface to the gd-library, provides easy access to library functions within TCL. The pbmplus tools [PBMPLUS 95] are used for format conversions.

3.3 GIS Functionality

A GIS has to provide detailed information of a geographic region. We split the GIS functionality into two parts: visualization of GIS data on the one, data manipulation and analysis on the other hand. Visualization is done by a server GIS, which is connected via CGI [McCool 94] to the WWW-server (see figure 2). Currently, we use GRASS. Further services, provided by individual GIS at the WWW server site, are to be provided by data transfer between these GIS and the server GIS. This enables the integration of several GIS methods, using a simple transfer interface between standard GIS data formats. Up to now, an offline interface to transfer data to the server GIS has been developed and implemented.

The architectural details for handling GIS data are shown in figure 2. We have WWW clients, the WAN, and the WWW server. Then, we have application level services, i.e., services for tabular data and GIS services. According to the overall architecture as described in Section 2, application level services use different system level services to fulfill their complex tasks. The latter can be divided into three groups:

1. services for different DBMS on different platforms as shown in the left branch of figure 2,
2. services for GIS data as shown in the right part of figure 2, and
3. services for visualization, which include extraction of maps, generation of business graphics, and a combination of both in single HTML result pages.

GIS services are further divided into services for geometric data and metadata. Complex geometric objects are stored in individual GIS formats, whereas metadata of each geometric object are typically stored in a relational DBMS. In our architecture, GIS data are imported into the server GIS. Hence, the server GIS can as well be seen as a buffer that stores data

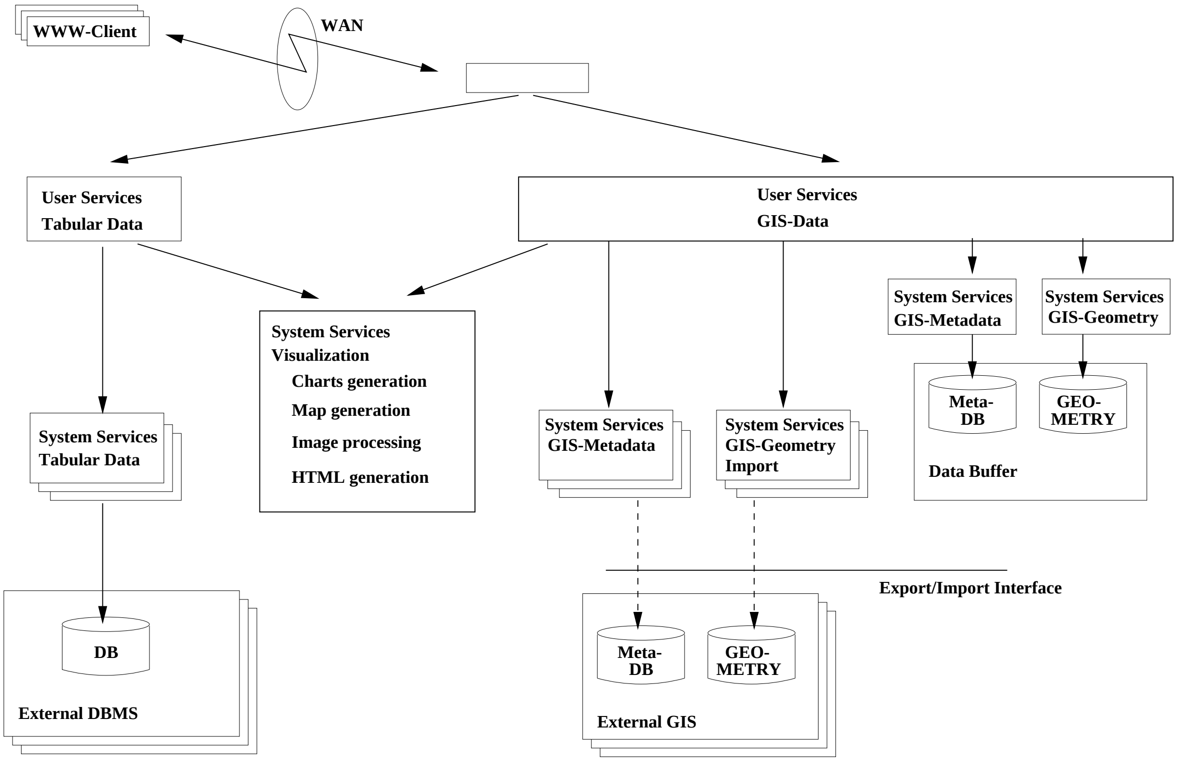

Figure 2: Architecture of GIS Integration

for a limited period of time. Since metadata and geometric data are closely interrelated, for obvious reasons, metadata is stored in this buffer as well. Thus, we have system services that provide data buffered locally to the user services and, on the other hand, services to import and export the data to several GIS. This approach makes both data and methods – originating in individual GIS – available on the Web.

Users can choose the region to be shown, different map layers to be displayed in a combination, and the style, in which the map and its elements appear in the final representation. Maps are separated into background and foreground maps. Background maps contain information for all points of the image. Foreground maps are any kind of vector or site maps. It is possible to combine one background with any number of foreground maps. Zoom facilities are provided for the maps. Furthermore, the user can mark a region and use this region for subsequent database queries, e.g., queries for UDK data.

3.4 Business Charts

To get familiar with the content of extended tabular data sets, a representation as business graphic is required. Since nearly all environmental data sets have a georeference, it is neccessary to visualize the georeference as well. Therefore, we have developed a method to combine business charts and maps. Charts may be displayed within the map or arranged around the map. An algorithm places diagramms such that lines do not intersect and that the overlay of diagrams within a map is reduced to a minimum.

To develop a tool for displaying diagrams, an interface for tabular data sets has been developed. Basically, query results are transfered to the visualization tool via files at the

WWW server site. These files contain the tabular data and an description of these, e.g., key lists for identifiers used within the tables. Each column of the tabular data has to be described. Furthermore, interchange formats have to be specified.

The user chooses a visualization goal from a presented goal list, and a parameter to be displayed on the abscissa. Additionally, the layout can be specified; users choose colors, the size of graphics, chart types (line, bar, pie), and the kind to combine them with the map. The graphics may be displayed as fullsize without the map background by a simple mouse click.

3.5 Example

The resulting HTML representation of a combined chart and map service is shown in figure 3. The map contains tabular data for different soil measurements in an industrial region near Heidelberg, Germany. The measurements are visualized as bar charts for different points on the map. The region displayed as a map is limited by the georeferences of the tabular data. The map shows different regions of the hierarchical administrative units: The line represents the town-area, different colors of the background map refer to the districts of the town. The site symbol of an airport is also included in the representation. Above the image, the symbols for the different operations are visible, like zoom or object information. After a symbol has been selected, the action is performed by clicking to a point of interest in the map.

4 A CORBA-based Middleware Layer

4.1 Motivation

WWW makes distributed information sources available. However, pure WWW technology does not support the combination of information from different distrubuted sources in a single HTML page. Information requests (queries) that access different databases or even combine a GIS (at location A) and a database access (at location B) are an example.

The introduction of a middleware layer is a solution to this problem. This layer is responsible for connections between different information sources (servers). Hence, data is still stored and managed at the place where it (organizational) belongs to. An example for data residing at different sources is UDK data [Lessing 96], e.g., one UDK residing in Austria and another one in Germany, each of them containing the country specific metadata. A query should now combine the results from both UDKs, giving a single result page. By using this concept, a federation of servers is built, as shown in the architecture of figure 1.

In order to build such federations of information sources and also to cope with the technical heterogeneity often found in reality, emerging technology standards can be used. An appropriate standard is the Common Object Request Broker Architecture (CORBA), which is described in the next Section 4.2.

4.2 Introducing CORBA

The Common Object Request Broker Architecture (CORBA) is a middleware layer that supports wrapping of distributed object providers (servers) and the use of these servers by clients [Object Management Group 95]. CORBA is defined by the Object Management Group (OMG), which now includes more than 570 members. In this architecture, which is defined independent from any hardware or software, a so-called object request broker (ORB)

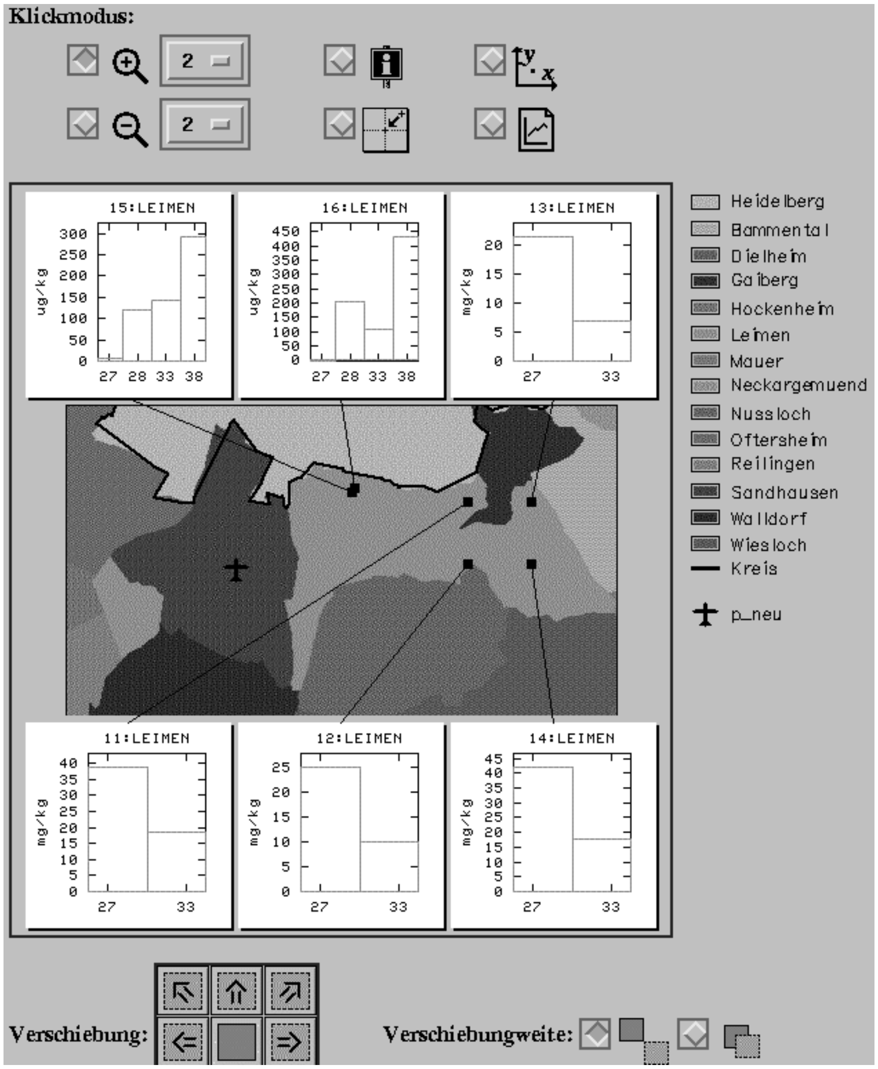

Figure 3: Example of GIS Representation in the World Wide Web

is responsible for distributing object calls made by clients and server responses to such calls. Important parts of the CORBA specification are a core object model, localization transparency (clients and server need not be aware of their respective locations, e.g. at different hosts), and programming language independence, realized by providing a specific interface definition language (IDL) as well as a dynamic invocation interface (DII) to objects. IDL is mapped to programming language specific bindings, e.g., for C and C++.

Currently, some 15 vendors offer CORBA implementations for different platforms, operating systems and networks. These range from PC systems or Macs using OS/2, Windows or MacOS, to workstations running UNIX and also mainframe environments.

4.3 Incorporating CORBA in the Federation Architecture

As shown in figure 1, the overall software architecture uses CORBA as the middleware layer, into which different system level services are integrated. Many of these system level services

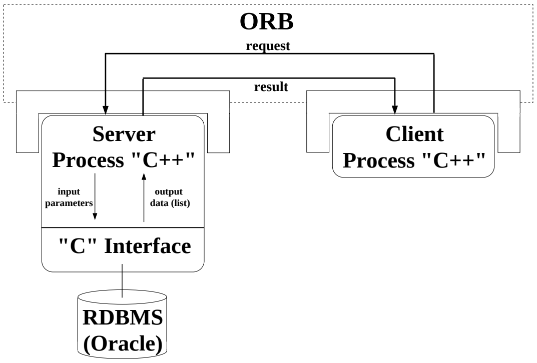

Figure 4: Integrating a relational DBMS into CORBA

(e.g., UDK-based information services, data services for limit values and measurement data) use a relational database system (RDBMS) to store their data. Thus, methods for accessing relational databases are important. We will give an example for this in Section 4.4.

Having developed a CORBA-based architecture, we are able to hide implementation details of services. In principle, these services can be used by any client application. Depending on the final graphical user interface, a client-specific preparation of results is necessary. In our case, we have a WWW server, which operates as a CORBA client. Thus, we provide WWW access to our system level services. Hence, results of system level services have to be prepared as HTML pages. We provide both a service-specific as well as a general preparation of these pages, which is beyond the scope of this paper.

4.4 Example

As an example for integrating a system level service using CORBA, we describe one approach for read access to RDBMS. The CORBA implementations ORBeline [Computing 94b, Computing 94a] and Orbix [IONA 93] have been used. The integration we developed works as follows. A CORBA server process, written in C++, has an IDL interface consisting of operations, which a client uses as database services. The operations in the server process call external C functions that access the RDBMS, e.g., Oracle. In particular, we access the data by using embedded SQL queries that are specified in the C functions. These C functions return a data structure to the server containing the results of the query. We use a linked list of structs as a fairly general though query specific structure for passing query results. In IDL, this linked list is defined as the CORBA predefined aggregation structure `sequence`, giving a kind of one-dimensional array of structs. The server is responsible for encapsulating this data and for passing the results back to the client for further processing. Figure 4 illustrates this process.

The structure information found in the IDL interface file reveals the structure of the data returned by the server. Based on this description, the client is able to manipulate the data as needed. He is responsible for deleting the `sequence` when it is no longer needed.

5 Related Work

In this section, we place our work in the context of GIS and the World Wide Web (Section 5.1) and accessing relational database systems from CORBA (Section 5.2). The list of approaches, concepts, and systems discussed does not claim to encompass all relevant work in the area. We restrict the discussion to those that we think to bear the most direct relationships to our problem and solution.

5.1 GIS and WWW

In a related project [Henning 96] ArcView is used at the client site to display maps. This method has the advantage to avoid processing of data at the server side, but additional software is necessary for every client. Our approach aims at a system which is accessible with a standard WWW browser. This enables the system to provide data and methods to everyone and avoid a lot of cost for large institutions using the data.

Some other scientists are working with GIS and WWW, most of them are situated in the United States. Often informations about the systems are only available on the web itself. Good starting points for a search are [U.S. Census Bureau 95a], [Thoen 95], [Behrens 95] and [OGIS 95]. Some examples of the capabilities of other systems are located at [U.S. Census Bureau 95b], [Huse 95], [Illinois Sate Museum 95] and [ERIN 95]. These systems offer different subsets of the functionality required in EIS. Our approach includes additional features like the combination of maps and graphics, as well as the intercommunication between mapping services and navigation systems as the UDK.

OGIS [Buehler 95] [Buehler (ed.) 94] aims at the definition of an universal, spatial and temporal data and process model. This model may be used for future integration tasks to solve many of the problems caused by heterogeneous GIS. The process of standardization is still going on, so a final solution of integrating GIS objects is not possible currently. Those standards enable an easy access of data and methods in heterogeneous GIS environments, but the style of a final map representation and the kind, datasets can be combined, changes for each individual GIS. In contrast, our approach uses a single GIS for data representation and combination, to enable a unique style and the creation of extended methods for data visualization.

5.2 Accessing RDBMS in CORBA

In several ongoing projects, CORBA is used to integrate RDBMS and other database systems. However, only few projects are reported in the literature. To given an impression of approaches, we refer to some of them.

In the EDS-Navigator-project [Quack 94] a control center for a power plant design and service system has been developed. An approach similar to our one is used to access the database.

For a World Wide customer information system [Hastings 94], a combination of WWW technology and CORBA has been used. An object wrapper layer has been developed, which passes information by pairs of name and value from HTML-forms to method calls of objects, as a generic approach to wrap information sources.

Jupiter [Grimson 94] and MIND [Dogac 95] are prototypes of multidatabase implementations that use CORBA as an interoperability layer. Such systems have to cope with arbitrary

(SQL) queries. Thus, a generic database system access-method had to be taken. This method uses generic access query interfaces of the underlying database systems, like a call-level interface, dynamic SQL or a SQL interpreter.

6 Conclusions and Future Work

In this paper, we have presented a federation architecture that provides the functionality of Geographic Information Systems using off-the-shelf WWW browsers. A description of further components of this system currently under development, e. g. online-reports about dangerous waste from the past, keyword-tools, in-depth evaluation of several CORBA implementations and Mosaic's common client interface (CCI) is to be found in [Henning 96].

No additional software tools at the client site are required. Furthermore, the systems supports access to relational database systems. Complex geometric GIS data and tabular data are visualized in maps combined with business graphics. CORBA is used to overcome some of the current limitations of the World Wide Web. The system is currently under evaluation in a private network; hence, we cannot provide an URL here.

Among the points to be addressed in the near future are the transfer of simple tasks from the server site to the client site in order to improve performance, and the integration of heterogeneous, distributed GIS methods and data. The first point will be addressed using JAVA. This new programming language offers a convenient way to transfer small application programs, so-called applets, to the client site. These applets will improve performance with respect to user interactions. Furthermore, they will reduce the transfer data volume by transfering vector data, which then will be rastered for the visualization at the client site. The second point is addressed using a CORBA-based Middleware as described in Section 4. With respect to CORBA and GIS, we carefully take into account the OGIS activities [Buehler (ed.) 94].

Acknowledgements: The students S. Behrens, U. Beutler, C. Habeck, P. Kirsch, B. Mayer, P. Palmer, C. Rolker, O. Scheffczyk, M. Schoel, and M. Walz were instrumental in the implementation effort at FZI. At IPF, the students M. Böse, C. Hofmann, and H. Krassin were involved in implementing GIS services and visualization tools. We gratefully acknowledge their support.

References

[Behrens 95] C. Behrens, C. S. Charles. Geosight project. Technical report, 1995. http://usdac2.rutgers.edu.

[Boutell 95] T. Boutell. Gd-library manual and sources. Technical report, 1995. ftp://isis.cshl.org/ pub/gd.

[Buehler (ed.) 94] K. Buehler (ed.). The Open Geodata Interoperability Specification. In *Draft base Document - OGIS Project Document 94-025R2*, Dec. 1994.

[Buehler 95] K. Buehler. The OGIS Technical Committee: Purpose, Status, Future. *Geo Info Systems*, page 48, Jan. 1995.

[Computing 94a] PostModern Computing. *ORBeline Reference Manual.* PostModern Computing Technologies, Inc., 1897 Landings Drive, Mountain View, California 94043 U.S.A, 1994.

[Computing 94b] PostModern Computing. *ORBeline User's Guide.* PostModern Computing Technologies, Inc., 1897 Landings Drive, Mountain View, California 94043 U.S.A, 1994.

[Dogac 95] Ebru Kilic, Gokhan Ozhan, Cevdet Dengi, Nihan Kesim, Pinar Koksal, Asuman Dogac. Experiences in Using CORBA for a Multidatabase Implementation. In *Proceedings of the 6th Int. Workshop and Conference on Database and Expertsystems Applications DEXA'95, London, UK.* Software Research and Development Center, Scientific and Technical Research Council of Turkiye, Middle East Technical University, Ankara, Turkiye, September 1995. e-mail: asuman@srdc.metu.edu.tr.

[ERIN 95] ERIN. Erin australia. Technical report, 1995. http://kaos.erin.gov.au/.

[Gleeson 95] M. Gleeson. Fly-tool manual and sources. Technical report, 1995. ftp://www.unimelb. edu.au/pub/cwis/tools/unix.

[GNU 95] GNU. GNUPLOT Manual and Sources. Technical report, 1995. ftp://ftp.dartmouth.edu/pub/gnuplot.

[GRASS 95] GRASS. GRASS users mailing list. Technical report, 1995. grassu-request@moon.cecer.army.mil, "HELP" within the mail body.

[Grimson 94] John Murphy, Jane Grimson. The Jupiter System: An Environment for Multidatabase Interoperability. Technical report, Dublin City University and Trinity College Dublin, Jan. 1994. e-mail: murphyjp@compapp.dcu.ie jane.grimson@cs.tcd.ie.

[Hastings 94] E. E. Hastings, D. H. Kumar. Providing Customers Information Using the WEB and CORBA. In *2'nd WWW-Conference*, 1994. http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/hastings/hastings.html.

[Henning 96] I. Henning, R. Mayer-Föll, M. Müller, E. Schmid, H. Spandl, A. Koschel, R. Kramer, W. F. Riekert, G. Wiest, W. Geiger, A. Jaeschke, R. Weidemann, F. Schmidt, J. Wiesel. *Projekt GLOBUS — Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg, Phase II - 1995.* Umweltministerium Baden-Württemberg, 1996.

[Huse 95] S. Huse. GrassLinks (A system based on the PD GIS GRASS). Technical report, 1995. http://www.regis.berkeley.edu:80/grasslinks/.

[Illinois Sate Museum 95] Illinois Sate Museum. Faunmap (based on ARC/INFO data). Technical report, 1995. http://www.museum.state.il.us/research/faunmap.

[IONA 93] IONA. Orbix – a technical overview. Technical Report PR-TEC-7-5, IONA Technologies Ltd., July 1993.

[Kramer 95] R. Kramer, H. Spandl. Metadatenzugriff in Weitverkehrsnetzen: Eine Realisierung am Beispiel des Umweltdatenkatalogs UDK. In F. Huber-Wäschle, H. Schauer,

P. Widmayer, editors, *Herausforderungen eines globalen Informationsverbundes für die Informatik; 25. GI-Jahrestagung und 13. Schweizer Informatikertag, Zürich, 18.-20. September 1995/GISI 95*, Informatik Aktuell, pages 610–617. Springer, 1995.

[Kramer 96] R. Kramer, T. Quellenberg. Global Access to Environmental Information. In R. Denzer, D. Russel, G. Schimak, editors, *Environmental Software Systems; Proceedings of the International Symposium on Environmental Software Systems, 1995*, International Federation for Information Processing (IFIP), pages 209 – 218, London, 1996. Chapman and Hall.

[Lessing 96] H. Lessing, W. Swoboda, O. Günther. UDK: A European Environmental Data Catalogue. In *Third International Conference/Workshop Integrating GIS and Environmental Modeling*, Santa Fe, New Mexico, USA, Jan. 1996.

[McCauley 96] J.D. McCauley, K.C.S. Navulur, B.A. Engel, R. Srinivasan. Serving GIS Data Through the World Wide Web. In *Third International Conference/Workshop Integrating GIS and Environmental Modeling*, Santa Fe, New Mexico, USA, Jan. 1996.

[McCool 94] Rob McCool. Common gateway interface overview. Technical report, National Center for Supercomputiong Applications (NCSA), http://hoohoo.ncsa.uiuc.edu/-cgi/overview.html, 1994.

[Object Management Group 95] Object Management Group. *The Common Object Request Broker: Architecture and Specification.* Object Management Group, Inc., 2.0 edition, July 1995.

[OGIS 95] OGIS. Open GIS Consortium. Technical report, 1995. http://www.ogis.org/-ogis.html.

[Ousterhout 94] John K. Ousterhout. *Tcl and the Tk toolkit.* Addison-Wesley, Reading, MA, 1994.

[PBMPLUS 95] PBMPLUS. Pbmplus manuals and sources. Technical report, 1995. ftp://sunsite.unc.edu:/pub/Linux/apps/graphics/convert.

[Quack 94] K. Quack. Corba-Implementierung macht Einzelanwendungen zum System. *Computerwoche*, Nov. 1994.

[Rodcay (ed.) 95] G. K. Rodcay (ed.). *GIS World Sourcebook 1996.* 1995.

[Thoen 95] B. Thoen. WEB - GIS. Technical report, Oct. 1995. http://www.gisnet.com/gis/webgis.html.

[U.S. Army Corps of Engineers 95] U.S. Army Corps of Engineers. GRASS Manuals and Sources. Technical report, 1995. ftp://moon.cecer.army.mil/pub/grass/grass4.1.

[U.S. Census Bureau 95a] U.S. Census Bureau. Tiger Mapping Service TMS. Technical report, 1995. http://wings.buffalo.edu/geoweb/related.html.

[U.S. Census Bureau 95b] U.S. Census Bureau. TMS (based on TIGER data). Technical report, 1995. http://tiger.census.gov/.