INTEGRATION OF GIS AS A COMPONENT IN FEDERATED INFORMATION SYSTEMS

Claus HOFMANN, Manuel WEINDORF, Joachim WIESEL
University of Karlsruhe, Germany
Institute of Photogrammetry and Remote Sensing
{claush,weindorf,wiesel}@ipf.uni-karlsruhe.de

KEY WORDS: GIS, Interoperability, Open Systems, Spatial databases, Spatial data infrastructure, Networks

ABSTRACT

Information systems are developing away from providing very specialized centralized solutions towards distributed systems based on the concept of services and objects. The main reasons for this new design paradigms are the possibility of coupling information systems to gain synergy effects, because many questions that need to be solved (e.g. in planning processes) are related to several different knowledge domains. Another reason is the possibility of spreading these information services into a broad public or user community by using internet techniques.
In many fields (e.g. environmental information systems, traffic information systems, tourist information systems) geospatial data plays (or could play) a major role. Thus the integration of different systems and data sources in practice is a challenging task of the GIS community, because at the moment there exists a multitude of different data formats and monolithic GIS architectures. Each of them uses different data models and provides proprietary tools to access, manage and visualize the data. This fact handicaps the interoperability between GIS as well as the integration of geodata and GIS components into other application domains.
This paper describes first structures and concepts of federated systems in general and further technical and implementation aspects of those component, middleware- and internet based techniques. To bring this concepts to practical applications at the IPF a special framework (named GIStermFramework) has been developed which offers generic GIS components that can be easily integrated into component based applications and internet based information system architectures.
It is shown that GIS can be integrated into federated system scenarios by using middleware- and standard internet techniques like Java/RMI, Java/InfoBus and HTTP/HTML. This possibility is illustrated through an environmetal information system (EIS), which shows that as a result of the cooperation of basically heterogeneous systems users have advantages while integrating different knowledge domains.

1 INTRODUCTION

During the past few decades Information technology has improved in many ways. This has lead to the development of several information systems in different disciplines and for different applications. In most cases these systems were developed in a very specific context aimed at the solution of a very specific question. Therefore problem oriented data models were defined and applications based on this models were implemented.

Meanwhile the development of the Internet as a universal communication medium with over 40 Million users (Buhmann and Wiesel, 1999) opens up a broad field of new chances and utilization possibilities. The role of the internet is getting more and more important for GIS. It is not just a medium for communication between provider and user but also initiates technical developments directly applicable (e.g. HTML, XML, Java, TCP/IP etc.).

Furthermore there is a change away from centralized systems towards distributed systems. This results in software which is easier to maintain from the technical point of view and from the users point of view, there is an improvement of scalability and adaptation to business processes as well as management structures. This evolution of information systems offers a basis for the integration of spatial data and components into all kinds of application domains.

Geodata is a fundamental and important part of the data warehouses of administrative or commercial organizations. Their integration into nearly all kinds of applications and workflows is one of the actual challenges of the GIS community. At the moment there exists a multitude of different data formats and monolithic GIS architectures. Each of them uses different data models and provides proprietary tools to access, manage and visualize the data. This fact handicaps the interoperability between GIS as well as the integration of geodata and GIS components into other application scenarios. On the other hand it prompts GIS vendors to change their products away from monolithic universal solutions towards component based architecture (Buhmann and Wiesel, 1999). This means for the system developer, that very specialized standard components are available and have to be combined into custom-made solutions.

In this way it is possible to provide as much GIS functionality as the (non expert) user needs to solve his/her day-by-day problems directly onto his/her desktop. GIS can be integrated into standard office workflow resp. applications. That leads to a broad usage of geospatial data because even non GIS-experts are able to access this data within a familiar environment. In order to the immense costs of geodata production (up to more than 80% of the overall costs of a GIS project (Buhmann and Wiesel, 1999)), this multiple usage presents a buissness advantagege. Planning processes often depend on more than one discipline, therefore it's very important to access actual and extensive data sources of other disciplines and to integrate these into one's own planning- and decision processes (Hosenfeld, 1999). Through a common use of services and informations, synergy effects can be obtained and quality of decision results should be improved.

2 GENERAL CONCEPTS OF FEDERATED SYSTEMS

A federated system consists of the combination of several autonomous systems (Lockemann et al., 1993), (Lang et al., 1995), which were developed and used independently and which serve several independent aims. In this section, some general concepts and features of such interoperable systems are explained.

Information systems generally provide a set of well defined functions which can be declared as services. Generally speaking these services are modules of functions which are demanded frequently by one or more service users and therefore are delegated to a special facility (Lockemann et al., 1993). A federated system uses services of different systems and integrates them into an higher level system. Thereby it's able to merge discipline spanning information to gain synergy between the connected information systems. The involved systems interoperate by the control of the superordinated system. There can be divided between two basic forms of interaction (Lockemann et al., 1993):

1. Cooperation
2. Coordination

In cooperation, two or more instances (systems or components) use the same common data source (or a part of it). On the other hand the coordination principle is based on copying the desired data between the scopes of the instances (see figure 1). Federated systems in general form distributed applications because of the independent development and the

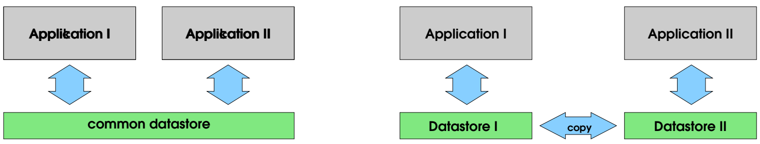

Figure 1: Cooperation vs. Coordination

different basic aims of the involved systems. Distributed applications consist of several processes which are running on distinct computers and which are operating together within a common application context. Dependent on the specific purpose of interaction these structures can be classified as follows:

Service or function based integration: This kind of integration is used to interact with domain specific application services (functional services). The aim is to build domain specific service repositories and reuse them in several application environments. For instance a map visualization service can offer a service that displays maps for named topics.

Data based integration: The structure of this type enables the integrated access to distributed and heterogeneous data sources from different computers. Based on this principles it's possible to combine data originally stored in different file- or database systems and process that data in an integrated way. One example is a GIS component that combines layers of different data sources in a map overlay.

Load based integration: A structure of this type aims towards a load balancing between the involved computer systems.

These different types of integration are frequently mixed whereby the combination of service and data integration is most usual in information systems. The freedom to mix services leads to another basic characteristic of federated systems. Those systems are based on interchangeable pieces of software (components). This offers the opportunity that services can be replaced by other, possibly faster and better, services without changing the applications code. This concept is called service interoperability, which describes the ability of applications to specify services that provide a special functionality. Interoperable services can be implemented with open software technologies which will be introduced in the next section.

3 STANDARDIZATION AND BASE TECHNOLOGIES

The aforementioned concepts of the latter section can be carried out on two conditions. First, system concepts have to comply with standardization efforts and second, system architectures and applications have to use open software technologies. Because of the scope of this article, the interoperable integration of GIS services into federated systems, this section first introduces actual standardization efforts of the GIS community. After that, important software technologies, which can be used to create federated architectures, are presented.

3.1 Standardization of GIS services

New or redesigned spatial system components have to comply with the standardization efforts of the OpenGIS (Open Geographical Information Systems) Consortium (OpenGIS Consortium Inc., 1998). This consortium is an open, industry-wide consortium of GIS vendors and users who are attempting to facilitate interoperability by proposing standards for GIS knowledge interchange. OpenGIS aims at the definition of an universal, spatial and temporal data and process model. The OpenGIS Simple Features Specification (Open GIS Consortium Inc., 1999) is the first available definition of an object model for GIS data and includes basic access operators and the definition of a simple geometry format.

3.2 Object oriented and component based technologies

Recent innovative developments in computer science can be used to implement open system structures and applications. Such an innovative base technology is Java (Gosling et al., 1996), which is an object-oriented programming language that has been designed especially to be used on networks of heterogeneous computer systems. Java can be used both, stand alone and in conjunction with the WWW (World Wide Web). Small Java programs, so-called Applets, are transferred as byte code from a web server to a browser using HTTP (HyperText Transmission Protocol, cf. 3.4). These applets are executed by a virtual machine that runs in the browser.

Furthermore Java defines a rich and useful component technology, called JavaBeans (Englander, 1997) and Enterprise JavaBeans (Denninger, 1996). This technology covers graphical and non-graphical components as well as components on the server side. The Java language and environment, including the JavaBeans specification, provide mechanisms for creation and management of software components whose functions represent the building blocks used in modern applications. The JavaBeans specification additionally provides powerful mechanisms, like events and bound properties for directly interconnecting components.

An extension of the JavaBeans component technology is the InfoBus (JavaSoft, 1998) architecture. This technology provides extensions to the standard JavaBean component communication mechanisms for dynamic data interchange and communication between components. The InfoBus allows application designers to create data flows between cooperating components. This interaction model is motivated by the communication mechanisms of physical components within a hardware bus. Multiple consumers can receive and use data published by a producer, and a consumer can obtain data easily from multiple producers.

3.3 Middleware technologies

Middleware technologies like Java RMI (Remote Method Invocation) (JavaSoft, 1996) or CORBA (Common Object Request Broker Architecture) (Orfali, 1997), (Sayegh, 1997) are instruments to build large, distributed and service based system architectures. Such federated architecture is necessary if components are distributed over several computers as well as distributed over separate organization units.

Java RMI is an integral part of the Java core and in consequence freely available. RMI deals with remote objects. A remote object is an object whose methods can be invoked from objects executed under the control of a different virtual machine (possibly running on another computer). Correspondingly, a remote interface defines the methods of a remote object, and a remote method invocation (RMI) is the action of invoking a remote object. This functionality requires a unique object reference to each of the objects for addressing them. A memory pointer is insufficient in this case because the object may reside on a different computer system or in a different process. Therefore in general this object reference consists of a computer network address, a communication port and a unique reference number. This object references have to be requested by a standard naming service which provides an object reference by passing an unique object name (or service description). If the network wide references of the concerned objects are determined, these objects are enabled to communicate with each other. The RMI runtime system, usually called the Object Request Broker (ORB), handles all the work which arises by the communication process in a way transparent to the system developer. The major tasks in this context are the marshalling, the network transport to the remote system and the unmarshalling of the requested data into the remote address space.

The Common Object Request Broker Architecture (CORBA) is also a middleware technology that supports wrapping of distributed objects. CORBA is defined by the Object Management Group (OMG), which includes more than 570 members. This architecture also uses a Object Request Broker (ORB) with similar functional behavior like RMI. But in contrast to Java RMI, CORBA is independent from a specific programming language. This is a main characteristic of CORBA and its realized by providing a specific interface definition language (IDL) as well as a dynamic invocation interface (DII, DSI) to objects. IDL is mapped to programming language specific bindings, e.g. for Java, C and C++. Currently several vendors offer CORBA implementations for different platforms, operating systems (Windows, MacOS, UNIX) and networks.

3.4 Web Technologies and Web Based Applications

The World Wide Web (WWW) provides transparent access to distributed and linked documents. The most common and most popular format to present information on the Web is the HyperText Markup Language (HTML), a standardized language for creating hypertext documents. The HyperText Transmission Protocol (HTTP) is the fundamental communication protocol of the entire internet. It is a simple stateless client/server protocol between a client component (e.g. as Netscape Communicator) and a Web server.

Due to the Java technology, a new class of interactive Web applications can be created by the integration of dynamic software components (Java applets) into HTML documents. These applets are loaded from the Web server that offers the embedding HTML document and executed by the Java machine that runs in the browser environment of the user. This mechanism offers the possibility to access and transport service components dynamically to a client computer. In a federated system scenario, Java applets can be loaded for specific application services on demand. The communication between the dynamically loaded components can be managed by the InfoBus technology. Furthermore, those Java components can additionaly use middleware technologies (Java RMI or CORBA) to communicate with other server sided components distributed over the internet.

Therefore the Web provides, in conjunction with modern techniques, an excellent technical infrastructure to build federated systems.

4 GIS FRAMEWORK ARCHITECTURE

To bring the concepts and technologies, introduced above, to practical applications, a special framework (named GIStermFramework) has been developed at the IPF. This framework offers generic GIS components which can be accessed by end users via the WWW and can be integrated into component based applications and Internet based information system architectures. Online access and fusion of multiple data sources on demand, is another focal point of the framework. Moreover, the framework is intended to provide a clear separation between application development and data sources. The GIStermFramework has a three tier system architecture shown in figure 2. The upper system component is called GISterm and is intended to run on the end-users computer. This client consists of a Graphical User Interface (GUI) software layer and the client side part of the data management layer. The system component in the middle represents the application server which is called GIStermServer. The application server runs on a Web-Host, ideally a server host that has a fast network connection to the desired data servers. The GIStermServer contains the second part of the management layer and also an abstraction layer which deals with several data sources. The abstraction layer implements a uniform and virtual data store for the upper parts of the architecture. Core components of the abstraction layer are the data source adapters. For each connectable data source a corresponding adapter component is required, making this particular kind of data source accessible to the abstraction layer. Underneath the GIStermServer (cf. figure 2) shows several geodata sources (server). Currently, adapter components exist for SmallWorld GIS and MapInfo SpatialWare servers. Additionally adapters for two simple data servers, developed at the IPF, are present. One is internally based on ESRI-Shapefiles and one uses a BLOB-technique (Binary Large OBjects) to store geometry attributes in an Oracle-RDMS. Beside these active data sources the framework also supports spatial file data that can be accessed from the client via a Web-Server (HTTP) or directly through the file system (right side in figure 2). The entire network interface between the client and the server component is implemented with Java RMI. If this distributed three tier constellation is not desired, the framework can also operate in a classical client/server fashion. In this case the client (GISterm) and the application server (GIStermServer) are executed in the same process space.

The GIStermFramework approach distinguishes between different kinds of data operations that are needed in GIS applications:

1. Lightweight or highly interactive operations, possibly with specific behaviors for particular applications. Examples are panning and zooming the map view, selecting geographic objects, displaying descriptive data of geographic objects, changing the drawing priority of layers or changing the visual appearance of geographic objects.

2. Heavyweight operations that use application specific algorithms. For instance, all kinds of path finding algorithms or numeric data aggregation processes.

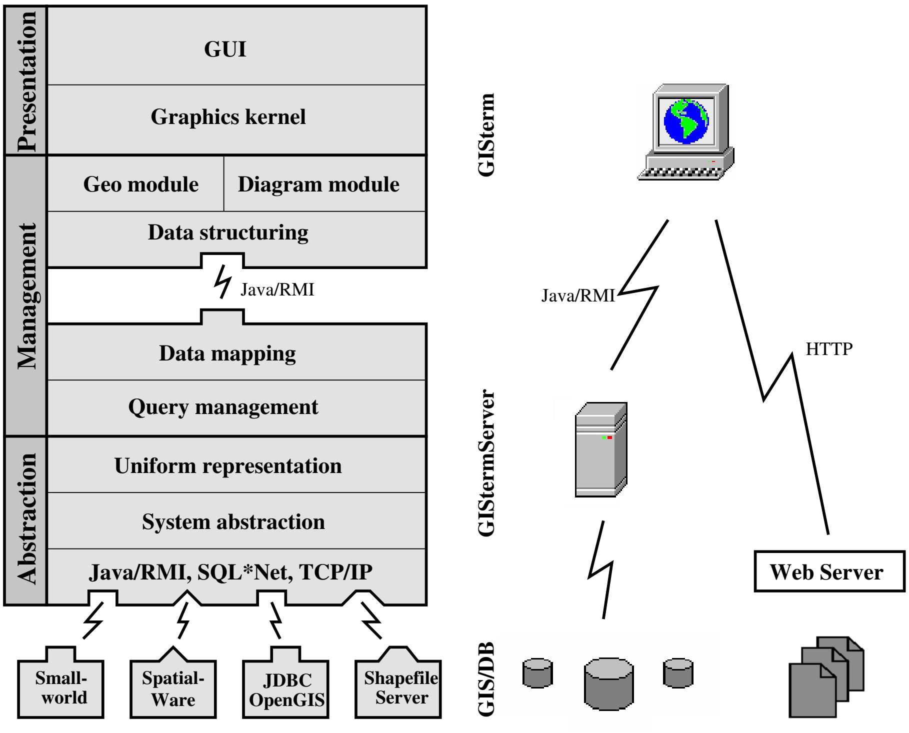

Figure 2: Three tier architecture

3. Generic heavyweight operations. This category covers mainly data selecting processes that can be formulated with descriptive spatial query languages (e.g. SQL with spatial extensions).

This classification is determined by the position in the system architecture where the operations should be implemented. Distributed GIS applications have to consider the fact of network capacity and load. The design of such systems has to comply with these limited resources. User acceptance of an application is influenced through the system response time. Users want immediate feedback to graphical interactive operations like selecting one or more geographic objects. In addition, all kinds of distributed applications have to minimize network traffic. Therefore, many lightweight and interactive operations should be implemented on client side. On the other hand, heavyweight and data extensive operations should be done on server side, to prevent unnecessary data transport. Only visual results of these operations should be transported to the client. To keep the framework independent of specific applications, these operations have to be implemented in an application server or in a generic database system. The GIStermFramework tries to hold a balance between client side operations and the costs of data transportation. Therefore the management software layer of the architecture is subdivided into a client and a server part. Further information on the internal design of the framework can be found in (Hofmann, 1999).

5 INTEGRATION IN A FEDERATED ENVIRONMENTAL INFORMATION SYSTEM

This section provide a practical example of a federated environmental information system and outlines the integration of interoperable GIS components into such scenarios. Another System, based on the same technology is described in (Weindorf et al., 1999).

In the last nine years an integrated Environmental Information System (EIS) has been developed in the State of Baden-Württemberg, Germany, managed by the Ministry of Environment and Transport (Mayer-Föll and Jaeschke, 1999). As technology has improved, the basic architecture of the EIS has been revised from a centralized to a distributed system based on the concept of services and objects. The main reason for this redesign was that environmental questions touch a large number of fairly different knowledge domains. These questions are normally addressed by different systems, which are usually heterogeneous (technically and semantically). The services of the EIS can be offered and used by various state agencies, which are connected through a computer intranetwork. To improve the accessibility to EIS by state officers and citizens, WWW has been chosen as the basic access and communication technology.

The following example describes a federated system scenario where three different services of the EIS are interacting within a web environment. This example is motivated by the information investigation that is necessary in an environmental planning process. The Following systems are involved:

1. The XfaWeb system (Weidemann et al., 1999), is a document retrieval systems that contains specific (e.g. geological or biological) reports from experts, workflow directives for environmental planning processes and other documents concerning administrative aspects of the environment. A user can search for documents via a theme hierarchy or using a keywords in a text pattern based search mechanism. The system is implemented with common web technologies. It uses HTML on client side to implement the user interface and web server extensions to implement the business logic on server side.

2. The SDS (attribute data system) is a highly advanced database retrieval system that gives access to several distributed databases. These databases contain descriptive data of environmental objects of the EIS (e.g. biotops, industry locations, online measurement values etc.). A user can choose several selection conditions for specific retrieval topics in an interactive manner. After query execution the resulting data can be presented by a several different ways. This includes a tabular view, the generation of paper reports and, via the GIS services, graphical presentations that visualize data inside a map or business chart. The system consists of a Java applet component that implements the GUI as well as the system logic. This applet also handles the query results in a local transient data store. A server part, also implemented in Java, is used to manage the distributed query processing.

3. The GISterm system gives access to spatial objects of the EIS. Therefore a user can choose geographic themes (provided by several geographic data sources of the EIS) from a unified thematic hierarchy and visualize them inside a map. Furthermore, one can explore the objects of an investigated area by a spatial navigation process as well as querying additional descriptive data about the spatial objects. The technical side of the system was introduced in section 4.

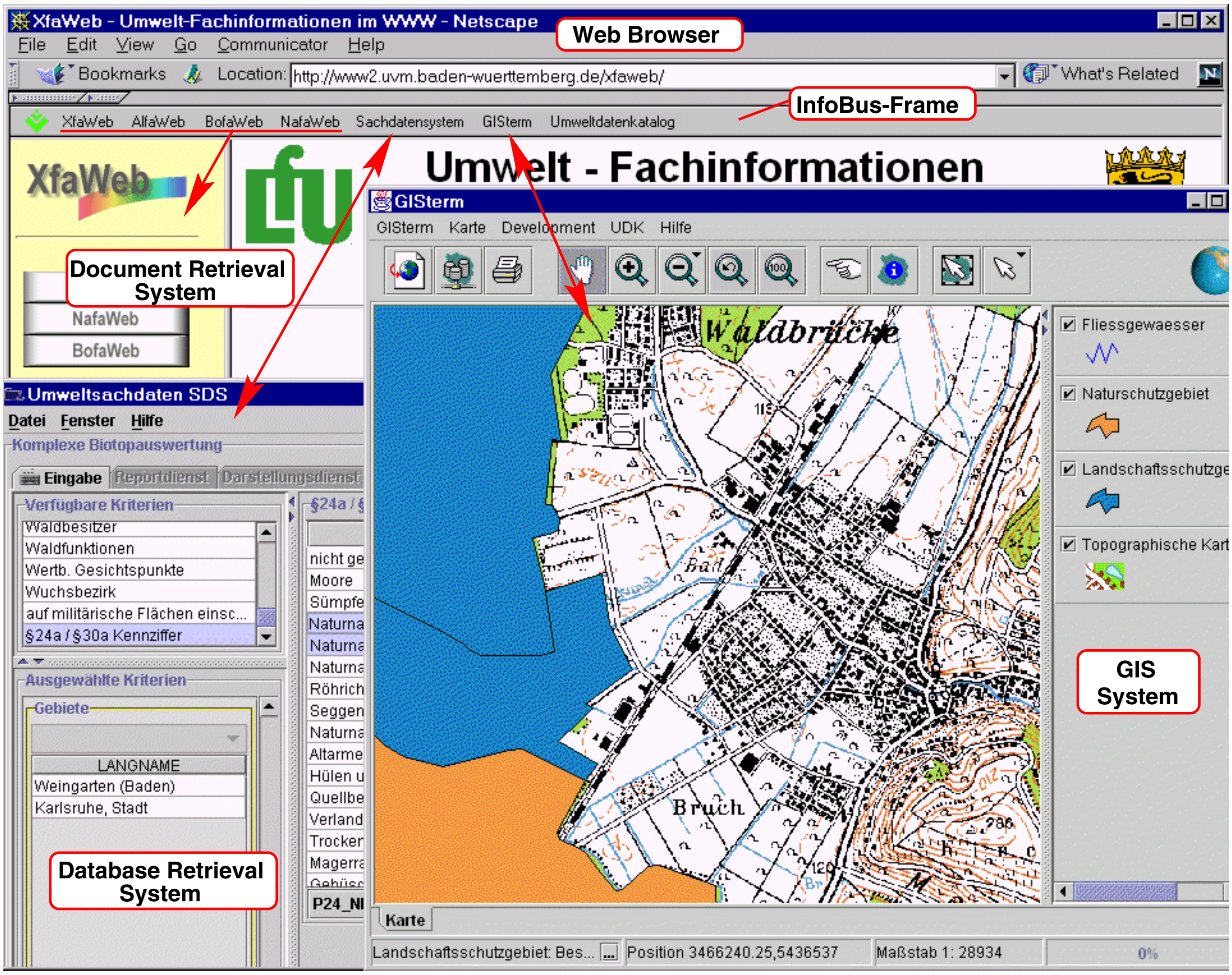

Figure 3: Screenshot of the EIS

These introduced systems, that work independent from each other, can be integrated into one federated system environment. To achieve that, we designed a web based portal to the entire EIS. Figure 3 gives a visual impression of the individual systems and the integrating web portal. This portal consists of a sophisticated (HTML-)frame layout that includes a frame for the Java applets (as well as the InfoBus) and additional frames for embedding HTML documents. All systems connect to the InfoBus and use this technology for interaction. Additionally, each Java client component opens a connection to their application server that is running on the web host. Figure 4 illustrates the architecture described above.

There are several advantages and new possibilities for the user of this federated approach. For example if a document provided by the XfaWeb system covers a special environmental object or theme (e.g. a biotope), this document can include a link that communicates with the InfoBus and connect thereby to services provided by the other systems. The GISterm, for instance, offers a service that integrates a new topic into a map by adding a new map layer to the current map. If the user activates the mentioned link he/she will see immediately all spatial objects of that geographical theme within the map of the actual area of interest. Another example: If a user selects attribute data within the SDS system and wants to visualize his/her results inside a map to recognize the spatial relationship of the resulting objects. The SDS can use another GISterm service to do that. This service uses the local and transient data store of the SDS to integrate this objects into a new map layer. This common data store is now used and managed by two different systems.

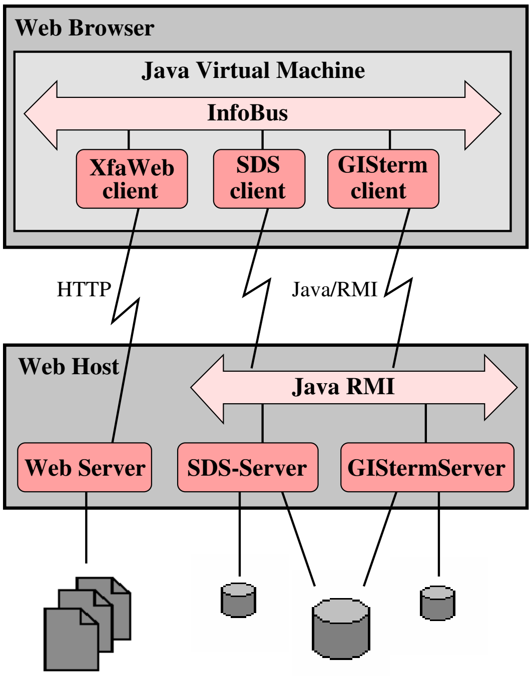

Figure 4: Federated EIS architecture

6 SUMMARY AND CONCLUSION

This paper has introduced structures and concepts of federated systems in general and further technical and implementation aspects of those component, middleware- and internet based techniques. Furthermore, the design of the GIStermFramework which offers generic GIS components that can be used within those modern system architectures was presented. Moreover, a practical example of a federated environmental information system that includes the introduced GIS components was given.

The paper gave an impression how federated systems can work together to support an information investigation process that is necessary for planning and management purposes in the future. Throughout the integrative use and the resulting synergies, the quality and the duration of a information retrieval process, covering several knowledge domains, obviously increases.

REFERENCES

Buhmann, E. and Wiesel, J., 1999. GIS Report 99. Bernhard Harzer Verlag, Karlsruhe.

Denninger, S. and Peters, I., 1996. Enterprise Java Beans. Addison Wesley.

Englander, R., 1997. Developing Java Beans. O'Reilly.

Gosling, J., Joy, B. and Steele, G., 1996. The Java Language Specification. Addison Wesley.

Hofmann, C., 1999. A multi tier framework for accessing distributed, heterogeneous spatial data in a federation based EIS. In: Proceedings of the 7th International Symposium on Advances in Geographic Information Systems (ACM-GIS '99), Kansas City, MO, USA.

Hosenfeld, F., 1999. Die Bedeutung von Informationssystemen für eine fachübergreifende ökologische Planung. In: Umweltmonitoring und Umweltmodellierung, H. Wichmann Verlag, Heidelberg.

OpenGIS Consortium Inc., 1998. Open Geodata Interoperability Specification. http://www.opengis.org.

JavaSoft, 1998. http://java.sun.com/beans/infobus.

JavaSoft, 1996. RMI documentation. http://java.sun.com/products/jdk/1.1/docs/guide/rmi/index.html.

Lang S. and Lockemann, P., 1995. Datenbankeinsatz. Springer Verlag.

Lockemann, P. C., Krüger, G. and Krumm, H., 1993. Telekommunikation und Datenhaltung. Hanser Verlag.

Mayer-Föll, R. and Jaeschke, A. (eds), 1999. Projekt GLOBUS – Von Komponenten zu vernetzten Systemen für die Nutzung globaler Umweltsachdaten im Umweltinformationssystem Baden-Württemberg und anderen fachübergreifenden Anwendungen, Phase VI. Ministerium für Umwelt und Verkehr Baden-Württemberg. http://www.uvm.baden-wuerttemberg.de/lfu/uis/globus/globus6/

Open GIS Consortium Inc., 1999. Opengis Simple Features Specification. http://www.opengis.org/techno/specs.htm.

Orfali, R. and Harkey, D., 1997. Client/Server Programming with Java and Corba. John Wiley & Sons, Inc.

Sayegh, A., 1997. CORBA – Standard, Spezifikationen, Entwicklung. O'Reilly.

Weidemann, R., Düpmeier, C., Geiger, W., Ch., G. and Zilly, G., 1999. Konsolidierung und weiterer Ausbau der XfaWeb-Systeme. http://www.uvm.baden-wuerttemberg.de/lfu/uis/globus/globus6/03-fzk/gl0603.html.

Weindorf, M. and Hofmann, C., 1999. GIS als Komponente in föderierten Systemen am Beispiel des "Informationssystems Binnenschiffahrt und Wasserstraßen", in Tagungsband 8 der DGPF: Photogrammetrie und Fernerkundung - Neue Sensoren, neue Anwendungen, 19. Jahrestagung der DGPF, Essen, 1999.