EnviroInfo 2000: Umweltinformatik ’00 Umweltinformation für Planung, Politik und Öffentlichkeit
Copyright 2000 Metropolis Verlag, Marburg, ISBN: 3-89518-307-5

A Component Based, Extensible Software Platform Supporting Interoperability of GIS Applications

Andreas Bergmann[1], Martin Breunig[2],
Armin B. Cremers[1] and Serge Shumilov[1]

Abstract

This paper summarises the experience gained in two projects carried out in the context of Interoperable Geo Information Systems at the University of Bonn. Evaluating the results of our research, we present the requirements for a general software platform supporting interoperability of Geo scientific tools and integrated processing of heterogeneous data. We also discuss the contributions of supporting technologies available today, such as CORBA, Java, and XML. Finally, we present our view of a flexible, extensible and adaptable architecture for an information infrastructure based on software component technology.

1. Introduction

Today a rapidly increasing volume of spatial and environmental information is collected using modern techniques. The resultant data is stored digitally in many different systems using heterogeneous data types and formats. These are defined according to local needs. The reuse of the data is often impossible due to missing metainformation like the information which is stored, its representation and structure, its quality, the date to which it refers, the scale employed, etc. This leads to widely isolated databases.

Moreover, many GIS and Geo scientific tools are traditionally closed applications that are not compatible with each other. Their reuse for new applications is a nightmare due to poor documentation, obscure semantics of data, heterogeneity in terms of data modelling concepts, data encoding techniques, access functionality, etc. Some vendors are starting to offer open interfaces based on standard “middle-ware” platforms like CORBA. For example, OpenSpirit provides a set of CORBA object services and specific objects to manage persistent interactive user sessions

[1] Institute of Computer Science III, University of Bonn, Roemerstr. 164, D-53117 Bonn, {andreas | abc | shumilov@informatik.uni-bonn.de
[2] Institute of Environmental Sciences, University of Vechta, P.O. Box 1553, D-49364 Vechta, mbreunig@ispa.uni-vechta.de

that use data servers that encapsulate data from a variety of data sources (Prism Tech. 2000). However, GIS technology still needs to make substantial headway in terms of interaction modes between different geo-scientific tools.

New applications in fields like telecommunications, facility management, catastrophe management, health care and environmental monitoring as well as research on advanced Geo scientific problems like classification of processes in different ecosystems would be of major benefit if the diverse and distributed information were readily available for integrated processing with adequate and highly developed tools.

In this paper we discuss the requirements for a software platform supporting interoperability of tools and integrated processing of heterogeneous data as well as the contributions of interoperability technologies available today, such as CORBA, Java, and XML. At the end we propose a flexibly extensible and adaptable architecture for an information infrastructure based on software component technology.

This paper summarises the experiences gained in two projects carried out in the context of the Interoperable Geo Information Systems (IOGIS)[3] initiative at the University of Bonn. Since 1996 six research groups from different disciplines, e.g. geography, geology, climatology, remote sensing and computer science have been examining different aspects of the integration of heterogeneous geo-scientific data and methods (IOGIS 1998).

The first project, called OPALIS[4], deals with uniform access to heterogeneous and distributed sources of paleoecologic data (Bergmann et al. 1999; Gärtner et al. 2000). Various German geo-scientific groups are participating in an effort to investigate the evolution of the biosphere during the last 15,000 years[5]. They are collecting large volumes of data describing local characteristics like strata of drillings, samples of pollen, results of chemical and physical analyses, etc. and storing them in many different formats and systems according to their particular needs. The characterisation and classification of processes in different ecosystems that will support detection of local particularities or anomalies according to superior characteristics and changes of the atmosphere can only be successful if the entire relevant information is accessible in an uniform way and can be processed with different tools.

The second project deals with 3D geological mapping (Breunig et al. 2000). The goal of the project is the construction of a consistent geological 3D model of an investigation area in southern Lower Saxony from primary geological data and its iterative refinement by alternate use of specialised geological - GOCAD[6] and

[3] This work is funded by the German Research Foundation (DFG)

[4] Open Paleoecological Information System (OPALIS)

[5] DFG priority program on Change in the Geo-Biosphere During the Last 15,000 years.

[6] GOCAD, a geological 3D modelling tool, http://www.ensg.u-nancy.fr/GOCAD/gocad.html.

geophysical - IGMAS[7] tools. Geophysical modelling employs gravimetric and magnetic evaluations of the potential fields to extrapolate the geological information gained at the earth's surface into the depth. However, in the initial stages it is not effective enough because of the large variability of parameters under consideration. To reduce this variability, the geo-scientific modelling needs a kind of roughcast, which can be provided by interactive geological modelling with GOCAD. The stratigraphic information obtained as a result of geological modelling is further used for the refined computations of densities within IGMAS.

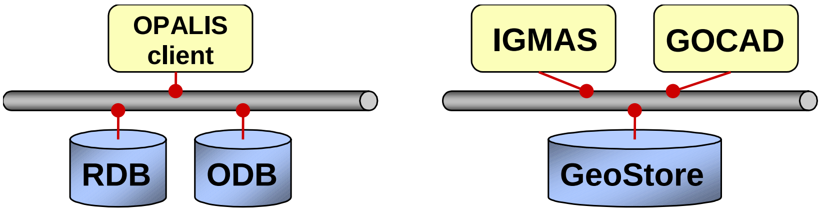

Figure 1:
Orthogonal interoperability aspects of the IOGIS projects

From the computer science point of view, the projects work on orthogonal aspects of the interoperability problem (Figure 1). In OPALIS, integrated access to multiple heterogeneous and distributed data sources and integrated processing of the a priori unknown data formats and types are to be supported in a client application. In the 3D geological mapping project, however, interoperability of two existing tools working on the same data in a distributed way has to be facilitated. This encompasses the detection of inconsistencies between the different views established by the tools.

2. Analysis of Requirements

2.1 Data Reuse and Exchange

Today's scenario can be outlined as follows. Usually, not only one Geo scientific application, but also varieties of applications running on heterogeneous hardware and software platforms are used for modelling, validating and visualising a specific area (Breunig et al. 2000). Therefore each of these applications needs to interact with the others through the exchange of data. The spatial data usually is stored in

[7] IGMAS, a geophysical 3D modelling tool,
http://userpage.fu-berlin.de/~sschmidt/Sabine_IGMAS.html

isolated way and is only accessible by their explicit owner. It will be maintained and accessed at different frequencies in different places, depending on current projects and ownership. Decentralised storage is recommended to ensure efficient availability of frequently used data at any time, but in order to break the isolation and promote data reuse and exchange all data should be made accessible in a uniform way transparent from its location.

Furthermore, the fact that data is stored in incompatible formats that places considerable demands on users and application programmers since they have to know the exact semantics of data. Often derived data cannot be interpreted without knowledge of the model used for its creation and the initial data used in the model. Any tool can handle no more than some of the data formats, and communication is only possible by means of exchange formats, leading to loss of semantics. As a consequence, only a small fraction of potentially available information can be exploited for today's challenging tasks in the context of spatial decision support and modelling of geo-processes like, for example, environmental monitoring and planning or the prediction of earthquake and climate evolution.

The Geo information system has to support identification of data meeting the users needs. In many current GIS solutions, no adequate user interface exists to guide users through available data and programs. Data access is only provided through predefined application programs and API with an embedded query language. Making data retrieval a programming task places high demands on the user, who is generally not assumed to be a computer specialist. It is an open question what information is appropriate in order to help Geo scientists find references to relevant data.

2.2 Extensibility with Respect to New Data Sources and Applications

Future GIS should not be monolithic and tightly integrated because there is a high risk that they cannot keep pace with changes in the rapidly emerging environment. The manifold examination methods used today in the different Geo disciplines deliver results that cannot seamlessly be integrated into today's GIS, because of their fixed data structures. Furthermore, expressions and analysis methods change over time, and newly discovered information has to be structured and included in the database. Due to the nature of research, the whole realm of spatial data will never be known. It is crucial to employ a data model that is flexible and extensible enough to deal with complex a priori unknown data types and can readily be adapted to future needs. Therefore one aspect of system extensibility will be a data model that supports addition of new types as well as the seamless integration of new modelling concepts. The second aspect of system extensibility is the ability to integrate and assimilate new pieces of software from different operational environments that may also be written in different programming languages. To achieve this goal the system should be composed of loosely coupled and easily maintainable units, controlled by a common infrastructure that provides mechanisms to ensure cooperation between those components.

Summarising all of the requirements presented above, the question arises as to what a geo-user really needs. Some fundamental points to be mentioned are:

- Seamless access to heterogeneous data sources and methods
- Support for identification of relevant data and computer based services
- Specialised and interoperable applications for different Geo scientific disciplines and problems
- Support for distributed cooperative research

3. Experienced Approaches and Technologies

Adequate interaction with different data sources that are connected to an information network requires effective and efficient mechanisms for selection of and access to relevant information at the data sites. Complementary components have to be provided to facilitate remote control of these mechanisms from client sites. Data is usually selected using a general-purpose standard query language or by means of an interface providing specialised functionality. Both approaches have rather complementary advantages and disadvantages with respect to our distributed scenario.

If a common query language is employed, the user has to learn only a single language in order to request data from different sources. Queries can also be formulated very flexibly within the boundaries of the respective operator combination rules. This approach poses some demands on the data sources. First, they have to export a schema describing the format of the maintained data that is expressed in terms of the underlying data model of the common query language. Furthermore, mapping between the common query language and the language used in the data source system is required in order to create equivalent queries for the actual data selection. This is a viable and useful approach particularly when a standard language like SQL or OQL is used at the data source. In this case the mapping is provided by software systems like ObjectDRIVER[8].

While traditional relational systems provide an appropriate technology for storage of many data types that are important in the context Geo applications, special database systems like GeoStore (Bode et al. 1994), which are based on extensible object oriented database technology, are required to effectively and efficiently maintain complex, spatially and temporally interrelated Geo objects. In order to benefit from the specialised functions and operators that will generally provide high performance compared to general purpose systems, an interface has to be available which provides all advanced options to the user in an appropriate way. The main

---

[8] ObjectDRIVER, an Open Object Wrapper dedicated to Relational Databases Reusing, http://www.inria.fr/cermics/dbteam/ObjectDriver

drawbacks of this approach are the fixed set of query functions. Furthermore, the user is forced to learn how to use the interface every time a new data source is connected.

We will go on to discuss the mechanisms that facilitate remote data access within the IOGIS projects. This work is part of a flexible, extensible, and distributed system architecture that is based entirely on software component technology (Szyperski 1998) like CORBA (Object Management Group 1999) and JavaBeans (Sun Inc. 2000). An overview of this architecture can be found in Bergmann et al. 2000. The use of an Object Request Broker (ORB) as the communication backbone enables both platform and programming language-independent implementation and transparent communication of the individual components by means of standardised protocols on the Internet (Shumilov & Breunig 2000). In order to attach new data sources to the information network, a wrapper component is needed in each case. Beyond the implementation of a small basic interface for identification and initial communication purposes, the wrapper has to provide adequate access to the data according to the heterogeneous requirements already mentioned.

3.1 CORBA/ODBMS Integration

We developed an alternative approach using our own experience gained during evaluation of commercially available systems to CORBA/ODBMS Integration. Compared with current OMG ideas for handling persistent objects, this approach pays more attention to support distribution for existing databases. The main idea is to extend the functionality of the CORBA standard object adapter for handling persistent objects stored in a database. The prototype of the adapter - eXtensible Database Adapter (XDA) was evaluated in the development of a CORBA wrapper for a real object-oriented database GeoStore (Shumilov & Cremers 2000).

In our study we tried to implement the most critical issues of CORBA/ODBMS integration that we missed in commercial adapters and paid more attention to supporting soft, non-intrusive CORBA integration for existing databases (Balovnev et al. 1998). It is very important for existing databases, since database schema evolution may be extremely time-consuming and any modifications in schema and applications are unacceptable. To prevent them XDA adds a layer of *transient intermediate communication components* that act as usual CORBA servers as well as database clients. In other words they *mediate* between CORBA client and database objects, converting data parameters and delegating all function calls in both directions. Because of this basic function we will name such components *mediators* (Wiederhold 1999). However, the price of this improvement is the responsibility of XDA to control the lifetime of mediators – their creation, deletion and binding to the corresponding database objects.

Construction of the wrapper around *GeoStore* shows that the use of XDA has substantially reduced the wrapper's development process and improves

CORBA/ODBMS system performance. The use of mediators not only saved database space, but also allowed us to make the integration without any changes in the original database schema and without disturbing existing local applications. The separation between the original database and CORBA components provided by XDA permits concurrent work with the database through both – CORBA and native C++ database communication interfaces. Moreover, when employing this method any already existing data store with a compatible schema can always be used through the CORBA interface. Referential integrity of the data shared by local and CORBA-conformant applications is guaranteed by the use of the native ObjectStore transaction control mechanism.

In the present distributed environment, the selected data may be processed remotely at the data source site or locally at the client site. The preferred location depends on properties of the system maintaining the data as well as application requirements. The first option is suitable for all operations involving the processing of large amounts of data. It is possible to significant speed up data processing if they are performed locally at the server site. This is especially interesting for web browser access to databases. With the integration of the Java language into the ORB environment, Java applets can interact with the persistent CORBA objects through domain-specific interfaces without any knowledge of how the objects are actually stored. At the same time, some small operations can also be processed at the client side. Read-only access to heterogeneously modelled information stored at distributed locations is to be supported in OPALIS, allowing for integrated data processing in the client application. This requires a generic data transfer mechanism serving as a base for mediation between the different data models of data sources and application components.

3.2 XML Based Data Transfer

The development of wrapper components is a very expensive task. In order to facilitate rapid connection for the large number of data sources that should be accessible in OPALIS, we tried to find techniques supporting a widely generic wrapper construction.

Our approach is based on object oriented modelling of the information stored in an underlying database by means of Object Definition Language (ODL) (ODMG 1999) and some extension like specification of units serving as semantic hints for future data integration. The resulting schema description is fed into a special compiler (Figure 2), that generates a set of different data-dependent software components readily available for construction of the respective wrapper. These components implement an exportable schema representation, an internal data model reflecting the schema description, interface definitions in CORBA IDL, and methods to convert the data into a generic XML format (W3C 2000).

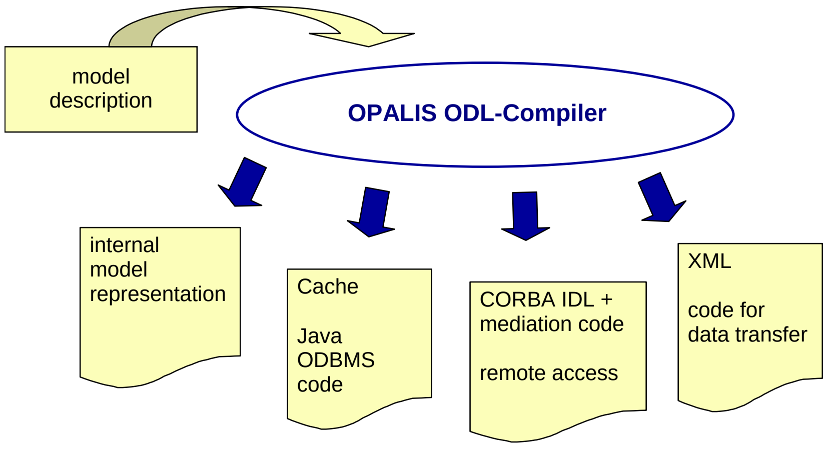

Figure 2:
Generated wrapper components

The XML format is defined by a document type definition (DTD) describing the object model. The inherent hierarchical structure of XML is mapped on the network structure of object models by means of the XLINK/XPOINTER languages. We can gain several advantages by using this technology for generic data representation and data transfer over the network. The data structure and semantic as defined by our model is fully available at any time and location. Due to flexible restructuring facilities (i.e. XSL) performance for data access can be optimised, on the one hand, and data can be formatted for processing with different tools on the other. All this can be achieved based on standard Internet technology and disposable tools, thereby opening the system to the web. Standard text compression algorithms as well as new compression technologies making use of the document structure can be used to further improve data transfer performance. Additionally, CORBA interfaces may be defined independent of concrete data structures, which in turn enables reuse of CORBA objects in broader contexts.

3.3 Data Navigation and Exploration

Data selection is supported by means of a portable graphical user interface in two different ways depending on the characteristics of the underlying data storage system. If OQL mapping is available as mentioned above, the data source can be queried through a generic user interface (Figure 3). Otherwise, a special purpose interface has to be implemented. Both kinds of interfaces should be implemented according to the principles of the OPALIS client platform. This platform was designed in order to support the diverse requirements of different users of the information network. Applications are viewed as flexibly configurable, user driven aggregations of specialised software components communicating through a communication component that facilitates the required local data exchange. A more detailed description of the platform architecture supporting “virtual” applications can be found in Bergmann et al 2000.

Figure 3:
OPALIS generic query interface

The generic query interface, which was developed as a part of the client platform, provides a list of all data sources connected to the information network. After the user selected a data source, the schema as well as the set of available object extents is displayed. The user may than formulate queries according to this information and send it to the data source for processing. When the result set is computed, the number of selected objects as well as the amount of bytes to be transferred to the client is shown. The user has the option to download the result set or edit the query to be more selective, for instance. The data can than seamlessly be processed by any other tool connected to the client platform.

4. Architecture for an Extensible Infrastructure

Summarising, we propose a general view of architecture. The following important features are required for this architecture:

- Standard-based architecture
- Intelligent communication with load balancing and customisable algorithms
- Advanced service administration tools & interfaces
- Designed to be highly scalable and support the integration of legacy systems
- Customisable & configurable, able to plug in own implementations
- Portable, reusable application code for maximum flexibility & productivity

A very important precondition in order to achieve the design goals stated above is the extensibility of the infrastructure. It should be flexible and configurable to meet the needs of different development and deployment scenarios. This permits developers to use those portions that suit their application needs and make extensions that exploit the infrastructure's services, though tailored to their specific application requirements. It covers the three common types of extensions that are facilitated by the architectural design of infrastructure:

- Extensions that allow additional data stores to be supported by existing data servers (data store specific source/sinks), addition of new data objects, attributes or methods to existing data stores.
- Extensions to allow data visualisation and processing at client side (a pool of portable Java components)
- Extensions adding new Geo scientific applications and visualisation tools

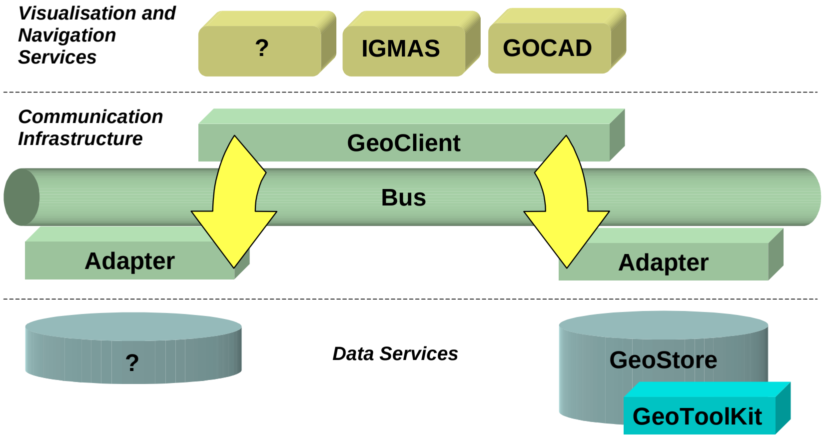

Figure 4:
Overall integration infrastructure.

Data Services: A Data Service is comprised of one or more data servers that provide create/read/update/delete access to common vendor data stores independent of the underlying data model. Where possible, the Data Service also includes an event gateway to the underlying data store and applications to translate vendor specific object selection, creation, deletion, and update events to/from standard events (Koschel 1999). Data Services also provide a convenient set of algorithmic building blocks for rapid application development. These components can take advantage of the Generalised Data Representation used in Data Modules. The Data Services use the Base Communication Infrastructure to provide a consistent environment for utilising the business objects contained in a Data Service.

Base Communication Infrastructure: The Base Communication Infrastructure should provide a set of object services and specific objects to manage persistent interactive user sessions that use data servers encapsulating data from a variety of data sources. It also defines communication interfaces and protocols that perform as a common interaction "language" between all components of the system. Conversion of units, geographic co-ordinates, and reference values are supported. The infrastructure is independent of any specific application domain and designed to support the full range of data services and client applications across the entire application domain and beyond.

Visualisation and Navigation Services: *Visualisation and navigation services* contain a set of visualisation or processing objects that can serve as building blocks for application developers. These will be introduced as optional additions to the common architecture.

Visualisation – 2D or 3D viewers – can display the business objects supported by the Data Services. These Viewers utilise the extensible client environment.

Navigation implements the query interfaces to support the workflow in a particular application domain. It provides a convenient set of visualisation building blocks for rapid development of custom query interfaces.

5. Conclusions

This paper presented a software platform supporting interoperability of tools and integrated processing of heterogeneous data. The results presented originated from the German IOGIS project of interoperable Geo scientific information systems. The design of the open architecture for an integration framework will serve as an extensible backbone for the localisation and integration of data sources as well as an infrastructure for inter-operating Geo-tools. The overall goal is to ensure the availability and consistency of data and its coordinated use and development by means of a completely integrated workflow of applications.

Although the discussion is focused on specific Geo scientific project research, we believe that the construction of complex distributed systems in many fields using

Geo Information will benefit fundamentally from the domain independent system architecture.

Acknowledgements

The authors would like to thank the German Research Foundation (DFG) for the financial support within the joint IOGIS project. This work would also have been impossible without the excellent cooperation with the groups of Richard Dikau and Agemar Siehl (both Bonn University) and H.-J. Götze (Free University Berlin).

References

Balovnev, O., Bergmann, A., Breunig, M., Cremers, A. B., Shumilov, S. (1998): A CORBA-based Approach to Data and Systems Integration for 3D Geoscientific Applications. In: Proceedings of the 8th Intern. Symposium on Spatial Data Handling, Vancouver, Canada, pp. 396-407

Bergmann, A., Breunig, M., Cremers, A. B., Shumilov, S. (2000): Towards an Interoperable Open GIS. In: Intern. Workshop on Emerging Technologies for Geographical Information Systems, Ascona, Switzerland, pp. 283-296

Bergmann, A., Gärtner, H., Breunig, M., Cremers, A.B., Dikau, R. (1999): Integrierter Zugriff auf heterogene paläoökologische Datenbestände. In: Rautenstrauch, C. and M. Schenk (eds.): Umweltinformatik 99. 13. Internationales Symposium Informatik für den Umweltschutz der Gesellschaft für Informatik (GI), Magdeburg 1999, Metropolis-Verlag, Marburg, pp. 403-416.

Bode T., Breunig M., Cremers A. B. (1994): First Experiences with GeoStore, an Information System for Geologically Defined Geometries. In: J. Nievergelt, Th. Roos, H.-J. Schek, P. Widmayer (eds.), IGIS'94 – Intern. Workshop on Advanced Research in Geographic Information Systems, Ascona, Switzerland, Lecture Notes in Computer Science, Springer, Berlin et al., pp. 35-44

Breunig, M., Cremers, A. B., Götze, H.-J., Schmidt, S., Seidemann, R., Shumilov, S., Siehl A. (2000): Geological Mapping based on 3D models using an Interoperable GIS. In: GIS Journal for Spatial Information and Decision Making, Vol. 13, No.2/2000, pp. 12-18.

Gärtner, H., Bergmann, A., Breunig, M., Cremers, A.B., Dikau, R. (2000): OPALIS - An Open Paleoecological Information System. In: GIS – Journal for Spatial Information and Decision Making, Vol. 13, No.2/2000, pp. 6-11.

IOGIS (1998): IOGIS – Vision einer neuen Generation interoperabler GIS. Bericht der teilnehmenden Gruppen der Universitäten Berlin, Bonn, Berlin/Bonn, Freiburg, Münster und Stuttgart, 13p.

Koschel A. (1999): Ereignisgetriebene CORBA-Dienste für heterogene, verteilte Informationssysteme. PhD thesis, Universität Karlsruhe.

Object Database Management Group (1999): The Object Database Standard: 3.0, R. Cattell (ed.), Morgan Kaufmann.

Object Management Group (1999): CORBA 3.0 New Components Chapters, OMG TC Document ptc/99-10-04.

Prism Technologies (2000): OpenSpirit / OpenFusion. http://www.prismtechnologies.com/.

Sun Inc. (2000): Enterprise JavaBeans™ 2.0 Specification.

Szyperski, C. (1998): Component Software – Beyond Object-Oriented Programming, Addison-Wesley, Essex, England, 411p.

Shumilov, S., Breunig, M. (2000): Integration of 3D Geoscientific Visualisation Tools. In: Proc. of the 6th EC-GI & GIS Workshop. The Spatial Information Society - Shaping the Future, Lyon, France.

Shumilov, S., Cremers, A. B. (2000): "eXtensible Database Adapter - a framework for CORBA/ODBMS integration", In: Proceedings of the 2nd International Workshop on Computer Science and Information Technologies (CSIT'00), Ufa, Russia.

W3C - World Wide Web Consortium (2000): Extensible Markup Language (XML) 1.0, Extensible Stylesheet Language (XSL) 1.0, XML Linking Language (XLink).

Wiederhold, G. (1999): "Mediation to Deal with Heterogeneous Data Sources", In: A. Vckovski et al. (eds.): Interoperating Geographic Information Systems, Proc. of the 2nd Int. Conf., Zurich, Switzerland, Lecture Notes in Computer Science, no. 1580, Springer, pp. 1-16.