Towards an Interoperable Open GIS

Andreas Bergmann, Martin Breunig, Armin B. Cremers, and Serge Shumilov

Institute of Computer Science III
University of Bonn
{andreas,martin,abc,shumilov}@cs.uni-bonn.de

Abstract. The number of geo-science applications has been ever increasing over the last decades. Most of them, however, do not provide required level of data and systems integration and tend to be architecturally closed, monolithic, and costly environments. In this paper we present our approach to design and development of an open component-based Geographical Information System (GIS) architecture. The overall goal is to make geo-data commonly accessible and usable for everyone and everywhere, by providing remote, prompt and effortless access to geo-data. The approach is evaluated by the development of a prototype. This contribution describes substantial components of the system, which supports the access to heterogeneous, distributed volume of data serving as an extensible communication backbone for inter-operating Geo-tools as well as an infrastructure for localization and integration of data sources. The presented Client/Server architecture supports extension in two dimensions: integration of new data sources as well as corresponding tools for their exploration and analysis. The benefits of the new approach for geo-scientists are manifold and related to an eased handling of data and the integration of geo-scientific methods, all resulting in better validated models and consequently in more accurate and plausible results.

1 Introduction

The number of applications using spatial or geographic data has been ever increasing over the last decades. However, GIS technology still needs to achieve a substantial progress in terms of interaction modes between different geo-scientific tools. Usually they are traditionally closed isolated applications that are not compatible with each other. Their reuse for new applications is a nightmare, due to poor documentation, obscure semantics of data, diversity of data sets (what information is stored, how it is represented and structured, what quality it has, which date it refers to, which scale is used, ...), heterogeneity of existing systems in terms of data modeling concepts, data encoding techniques, storage structures, access functionality, etc. In order to overcome the resulting data isolation and to support advanced geo scientific investigations based on heterogeneous data we propose an approach to an open component-based Geographical Information System (GIS) architecture. Since 1996 six research groups from different disciplines, e.g. geography, geology, climatology, remote sensing and computer science are working in the context of the "Interoperable

Geo Information Systems" (IOGIS)[1] initiative. The projects examine different aspects concerning the integration of heterogeneous geo-scientific data and methods (IOGIS 1998). The work described here was carried out by two subprojects of IOGIS at the University of Bonn.

The first one deals with uniform access to heterogeneous and distributed sources of paleoecologic data (Bergmann et al., 1998; Bergmann et al., 1999). Multiple German geo-scientific groups participate in an effort to investigate the evolution of the biosphere during the last 15.000 years. They collect lots of data describing local characteristics like strata of drillings, samples of pollen, results of chemical and physical analyses, etc. and store them in many different formats and systems according to their special needs. The characterization 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 whole amount of 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., 1998). The goal of the project is the construction of a consistent geological 3D-model of Southern Lower Saxony from primary geological data and its iterative refinement by alternate use of specialized geological - GOCAD[2] and geophysical - IGMAS[3] tools. Geophysical modeling applies gravimetric and magnetic evaluations of the potential fields to extrapolate the geological information gained at the earth surface into the depth. However, on the initial stages it is not effective enough because of the large variability of parameters under consideration. To reduce the variability the geo-scientific modeling needs a kind of roughcast, which can be provided by the interactive geological modeling with GOCAD. The stratigraphic information obtained in the result of geological modeling is further used for the refined computations of densities within IGMAS.

2 Motivation

Identification of appropriate data sources is an important precondition in order to overcome data isolation, as well as the support of adequate access to those data sources. A first significant step towards availability and accessibility of a huge amount of heterogeneous data was the development of databases and respective query systems, where large data sets are stored and maintained centrally using a well known homogeneous format. A major disadvantage of this approach, however, is the given structure of the central database. Every data provider has to transform his own model of data storage according to the central data structure. This leads to loss of semantics, because in most cases important details are not reflected in the central schema. Furthermore, the availability of new data depends on the frequency of data imports at the central site.

---

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

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

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

The IOGIS approach to solving the problem of data heterogeneity and accessibility is to represent the isolated data pools using object-oriented modeling techniques and to facilitate integrated access to heterogeneous data via the Internet utilizing the extracted meta-information. The goal is to leave every data set unchanged at its original site, not at least to guarantee a constant actuality of available data. An object model describing the structure and semantics of the exportable portion of the data is developed to facilitate the integration of the respective data source into the data network.

Merging all local schemas into a global schema (common approach) is not a satisfactory way of data integration, because the semantic richness of the data could not be modeled in detail (Abel et al., 1998). IOGIS uses a two level approach of data representation instead. At the first and more abstract level of information, the data of each connected source is mapped on a common object model that represents fundamental meta-information on geo-scientific data. The user may query this information in order to identify data sources that comprise relevant data in his project context. Once provided with a set of data sources, a user may, at the second level of information, directly interact with a connected data site by means of the respective object model and a generic data transfer protocol. This facilitates accessing every exported portion of a data set at any level of detail without any loss of information. The design autonomy and local maintenance of the data according to special requirements is preserved.

3 IOGIS architecture

In order to support integrated access to distributed heterogeneous data sources over the Internet, an open and extensible system is developed. The main design goals are on one hand easy integration of new data sources and on the other hand, the possibility for users to explore the great diversity of data types with adequate computer based tools.

To achieve the required flexibility and extensibility, we propose an object-oriented architecture, that is completely based on software component technology (Szyperski, 1998) like CORBA (Object Management Group, 1999) and JavaBeans (SUN, 1997) and uses a generic protocol to transfer and interact with data. Thus the whole system can be assembled from independent exchangeable components on different granularity levels. CORBA middleware was used to support the distributed access to heterogeneous software systems. The use of Object Request Broker (ORB) as communication infrastructure enables both platform and programming language-independent implementation and transparent communication of the individual components by means of standardized protocols on Internet (Balovnev et al., 1998). Data is communicated within IOGIS using a generic object format. This is represented as needed either by instantiating an object or using a textual representation based on the extended markup language (XML) (W3C, 1998). In the following we give an overview of the most important building blocks of the system, as shown in Fig. 1.

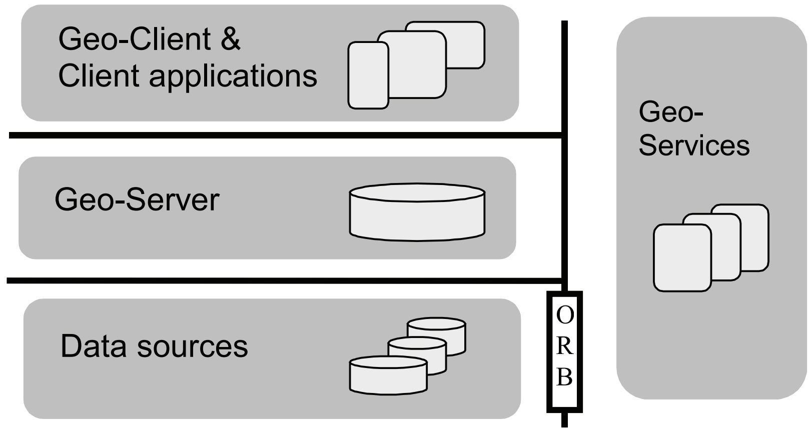

Fig. 1. Main components of the IOGIS system architecture

The data sources depict the sites that are connected to the information network. Due to the diversity of employed systems, computer platforms, and data formats, a wrapper component realizing the data source interface is needed for every type of data source. In order to support implementation of such component, a framework helping to minimize realization cost for the system dependent part is developed. To adequately support realization of the data dependent part with respect to universality and efficiency, a compilation-based approach was chosen.

The main task of the **Geo-Server** is to support the identification of relevant data within the connected data sources with respect to certain categories like space, time, or thematic data as described in the meta data model. Take for example the following query: "which data sources provide data in a given rectangular area?" Because connected data sources may model space in different ways the server has to utilize meta-knowledge to apply appropriate transformations like coordinate conversion or utilization of a map grid. Geo-Server provides a query facility for integrated meta-information and an appropriate exploration of the concrete raw data.

The **Geo-Services** are an extensible set of components providing different geo-scientific methods or data services. They are accessible by all other components via remote method calls. The Geo-Server for instance uses them to compute the necessary transformations. At present, some functions to the conversion of units and different cartographic projections are available. The service components are also available for use from client applications. Here, however, the primary topic of the interest is access to geo-scientific methods and their reuse in the new client applications.

The client applications are geo-scientific applications, which can be already used or developed on special needs by some researchers. It is not necessarily a new specially developed application, rather there can be an already existing application, e.g. GOCAD and IGMAS. Among other things the user of such applications receive the possibility to exchange an information with each other - common extensible components-based **Geo-Client** provides an interactive exploration of existed data and its conversion to local application-dependent format.

4 The Geo-Server component

Generally, the responsibility of Geo-Server is to find an answer to the following question - what data are available, which are relevant for a request, who has that data and how can it be made available. There are two different strategies for data integration - the federated database systems (Conrad 1997) and data warehouse systems (Labio et al., 1997). Federated database systems assume that all data are stored locally in the source systems. A central server translates queries and passes them on to the data sources. Their responses are then collected and integrated by the server according to a global pattern. Therefore, the actuality of the received information can be guaranteed. The drawbacks of this method are possibly long response times and incomplete information, since individual data sources may be not available at query processing time.

The data warehouse introduces an opposite strategy. The necessary data is already integrated before an inquiry and stored in a central place. A user query does not have to be transformed and shipped to the original sources for execution. This can lead to significant timesavings, if frequently requested information is accessed. However, a potential drawback of the warehousing approach is that data is physically copied from the original sources. This leads to subsequent problems: necessity of *extra storage space* and probability of *data inconsistencies* in sources and in warehouse. Furthermore, only information that was considered at design time of the warehouse may be accessed this way.

In IOGIS we developed a mixed approach. Due to the huge amount of data and certain access restrictions, a large part of the data cannot be copied and maintained centrally. At the same time, the central maintenance of the data gathered and structured according to the requirements of a particular project is not acceptable due to availability and flexibility issues. On the other hand, a purely distributed architecture would lead to intolerable long waiting periods. First of all, this argument applies to the meta-information described above. It changes much less frequently as the real data and should be stored in the central warehouse.

Considering performance issues, storage of the real data in the data warehouse is advantageous. Initially located at the remote sources, the data could be promoted to the warehouse, when it is accessed for the first time. The central storage can be seen as a *materialized view* over the data in the individual sources and should act like a cache. This again leads to additional difficulties: one problem is *selection* of the data that should be materialized. In GIS we have to deal with multidimensional data and consequently multidimensional materialized views in the warehouse. Because of multidimensional data specific, common selection algorithms do not work well. Several techniques have been proposed in the past to perform the selection of materialized views for databases with a reduced number of dimensions. Nevertheless, when the number and complexity of dimensions increase, they are also not acceptable. Further investigation of efficient selection techniques like (Baralis et al., 1997), based on reduction of the solution space by considering only the relevant elements of the multidimensional lattice, is necessary. A related problem is the development of an adequate replacement strategy.

Our work concerning the Geo-Server system component initially aims at defining and implementing adequate semantic access structures, that will support queries on interesting categories in the geo-scientific context such as space, time and special domain-oriented terms. Discussions with different geo-scientific research groups lead to detailed information about both, the objectives of the research and meta-data corresponding to the data sets that were made available. For getting a precise overview, it is essential to document all information about the objectives of the project, associated institutions, persons dealing with data and the equipment used. Furthermore, one has to identify the different types of data, the methods used for data acquisition, processing and storage as well as interpretations of analyzed data (Gärtner et al., 2000).

To support, for instance, the identification of sources comprising data located in a certain area, the Geo-Server maintains a spatial index structure. Because the term "space" is modeled differently in general, knowledge about conversions of coordinate projections or about how a certain grid maps to coordinates is utilized for the maintenance process. The modeling of "time" is even more complex. The aim is to gradually enhance query options, as additional models become available. This ongoing work will lead to a meta-level integrated view of the connected data sources.

5 Integration of new data sources

To promote interoperability and support a seamless information flow from heterogeneous distributed data sources to diverse user and geo-discipline dependent tools we have to deal with mainly two dimensions of extensibility: connection of new data sources to the information network and a possibility to enhance existing applications to adequately deal with previously unknown, generally complex data types. We will now focus on our approach to integration of data sources. The description of extensible client applications is deferred to the next chapter

In order to attach a new data source to the data warehouse, a wrapper component must be implemented in each case. The wrapper represents a query component that translates a query from the internal query representation to the local query language, executes the query and passes the result back to the server. As an internal query language we use the Object Query Language (OQL) defined by ODMG consortium (Cattell, Barry 1997). The realization of the wrapper component depends on the particular system of the data source. Relational databases are frequently used tools for the management of data. This type of a source system we attach to IOGIS with help of ObjectDRIVER[4].

Geo-Server object-oriented database was made available to the CORBA environment with help of a special wrapper, which covers the native ObjectStore's C++ programming interface of the database with a new CORBA compatible IDL interface. The wrapper was developed on top of an eXtensible Database Adapter (XDA) – an object adapter for objects stored in a database.

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

5.1 CORBA/OODBMS Integration

A primary strength of Object-Oriented Databases (OODB) lies in their ability to model complex objects and inter-relationships among them. In the common case of an ODBMS that adds database features to the C++ like ObjectStore, database entities are instances of usual C++ classes. In addition to their state, they have also methods. To make them available to the clients as usual CORBA objects, a wrapper should wrap the interface of every database class by an equivalent CORBA compatible class. This class implements all their methods by the equivalents from the database class and is responsible for the correct data mapping between CORBA C++ and ObjectStore C++ data models. The instances of these classes act as usual CORBA objects as well as database clients. In other words they *mediate* between CORBA skeletons and database objects converting data parameters and delegating all function calls in the both directions. Because of this basic function we will name such CORBA objects *mediators* (Wiederhold 1997).

A database schema evolution may be extremely time-consuming, therefore the CORBA-compliance should be achievable without changing where and how the data are stored. It could be achieved only if the mediators are transient and an original database do not have any additional communication counterparts. The price of this is a responsibility to control the life circle of mediators – their creation and deletion. Care should be taken to avoid the creation of multiple mediators tied with the same database object (Amirbekyan 1999). Since standard CORBA object adapter – Basic Object Adapter (BOA) does not provide these features, we developed our own - an eXtensible Database Adapter (XDA) - an object adapter for objects stored in a database. The XDA does not completely replace the BOA, but represents its extension. Unlike the BOA, the XDA controls the life circle of mediators and deal with typical database features, e.g. locks and transactions. The XDA provides a CORBA client with an IDL interface for the native ObjectStore functionality allowing the client to control the management of persistent objects. The interface includes methods for manipulation with usual ObjectStore objects such as databases, segments, objects and transactions.

The XDA prototype is an almost complete adapter for the ObjectStore database system, providing wrapper developers with the full power of ObjectStore to define, manipulate, and share important application data. Through corresponding mediators XDA makes original ObjectStore classes (e.g. os_database, os_transaction) accessible for a CORBA client. However, often a database has its own classes that encapsulate the original ObjectStore classes and extend their functionality. In this situation XDA's extensibility allows a server programmer easily to redefine some parts of the native XDA functionality if it is necessary. The other XDA advantage is ready to use templates for user defined mediators. For every new database class a server programmer should implement corresponding mediator class. The reuse of mediator's templates provided by XDA drastically reduces the amount of the server code compared to other techniques and accelerates the development process.

In the same time the realization of a wrapper component depends on the data types and formats stored in the source system. In order to be able to integrate the data from different source systems, the semantics of the particular data must be described. For this purpose an object-oriented model is developed to express the schema of the respective data set using an extended ODL notation. The extensions of ODL are designed to provide semantic hints for the data integration process. By now type expressions may be annotated with an unit specification to support compatibility checks as well as securing of comparability by way of adequate conversion service utilization. An ODL compiler was developed, that processes a description of an object schema and generates required data dependant components fit for service within the XDA framework.

6 Extensible component based client

In order to support an adequate and detailed view on the connected data sources, instead of a fixed set of client applications or user interfaces a dynamically extensible *Geo-Client* platform is developed. It allows dynamic assembly of new plug-in components according to the user's requirements. Starting with a minimal *Geo-Client* application, the user may get more specialized components from the *Geo-Server*, as he deals with different data. The components may be downloaded explicitly on user request or implicitly along with suitable data. This approach has several advantages. Besides the direct development support for geo-applications, the user may immediately benefit from new or improved components, as they become available. Furthermore, applications can be easily enhanced to handle future data types and methods.

The plug-in components differ by their application level from the complex general components to simple specialized components. In the example presented in the next chapter you can find a complex component - configurable map, that can be used in different contexts and a simple component – a tool, which can be used for the visualization of drilling data, if certain stratigraphic information is present (fig. 2). The lower level components have a generic structure, but simple application, e.g. the visualization of data of concrete type in the map component. Such low-level components can be loaded if necessary transparently for the user, e.g. if a result of executed query contain data of new type, a corresponding visualization component will be found and automatically loaded into the client application.

Platform independence is a very important issue in the heterogeneous environment. The components must be deployable at any node in the computer network and communication should be location independent. Data must also be represented in a platform neutral way. Furthermore, a potentially growing system must also be scalable. To achieve these goals, an ORB (OMG, 1999) is used for communication between the components. In addition, most components are implemented using the Java platform (SUN, 1999) that makes them independent from concrete hardware platform. Implemented in the form of Java applet Geo-Client could be also loaded and used from the Internet browser, e.g. Netscape or Internet Explorer. This is especially important to facilitate the dynamic client approach in many environments.

A further important precondition in order to support flexible and dynamically extensible client applications is a generic and platform neutral data representation. Because of limited hardware resources at average client sites, the client application should not grow with every new data type sent from the server. Furthermore, data has to be communicated from the server to the client and between the components of the possibly changing client application. To overcome these problems, data is represented in a generic object format. By way of introspection, any component can deal adequately with every data object. The data can be interchangeably instantiated as a Java object implementing the generic object format or as a XML document. Currently we are using the XML format for data serialization and structure preserving, platform neutral representation for information transfer over the network.

7 A sample Geo-Client user session

When a user starts a Geo-Client application, he is first provided with the client integrator component. Here he may download new application components, and start or remove existing ones. The state of the client application is preserved between different sessions.

Fig. 2 shows a sample Geo-Client. The button column on the left side of the client integrator window shows two existing components, a digital map (inactive) and a visual query component (started). This query component supports direct access to a connected data source. The schema as described by the corresponding object model is shown, after a data source is selected. The user may create a query and send it via the Geo-Server to the data source. The example query is: "find all drillings with a depth more than 40 meters."

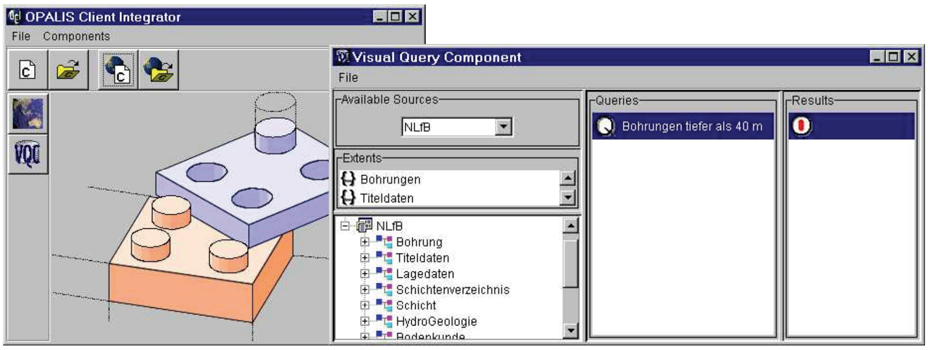

Fig. 2. Sample Geo-Client application –request for drilling data.

If the client is visiting a data site for the first time, the client software does possibly not know, how to deal with the special format for drillings provided by this source. However, the data may only be of full value to the user, if he can inspect it in any detail. Fig. 3 shows what happens after receiving the resulting drilling data.

Because the Geo-Client client is designed to handle arbitrary data types, there is no built in method to display this concrete drilling type. This problem is solved by extending the client with the most convenient method found in the component pool on the Geo-Server. This procedure is hidden from the user. In the example three icons corresponding to three different drilling types found at the data source are added to the legend and used to visualize the result data on the map. The user may configure the display method in order to adapt the appearance to his special needs.

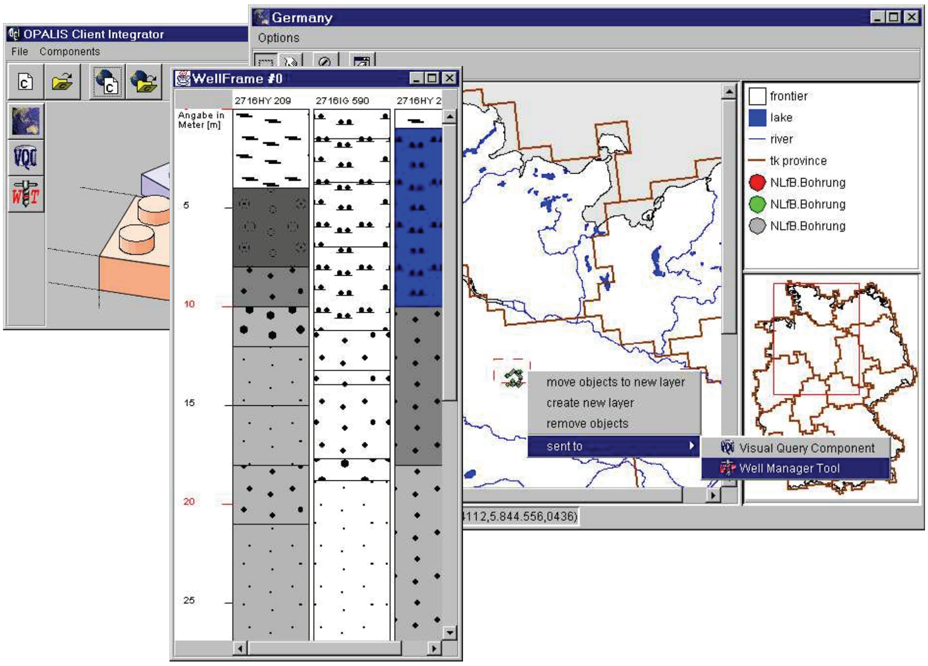

Fig. 3. Sample Geo-Client application – visualization of drilling data.

Furthermore, a new application component, the "well manager tool", may be integrated into the Geo-Client client at runtime. The corresponding start button appears in the client integrator window. This tool supports for instance exploration of stratigraphic information as shown in the "wellframe" window. Again, specialized display methods are used to visualize the actual data type.

8 An integration of data exploration and analysis tools

Another target of IOGIS architecture is the integration of data exploration and analysis tools during access to common integrated data sources. Here we present an example of integration between two 3D modeling tools.

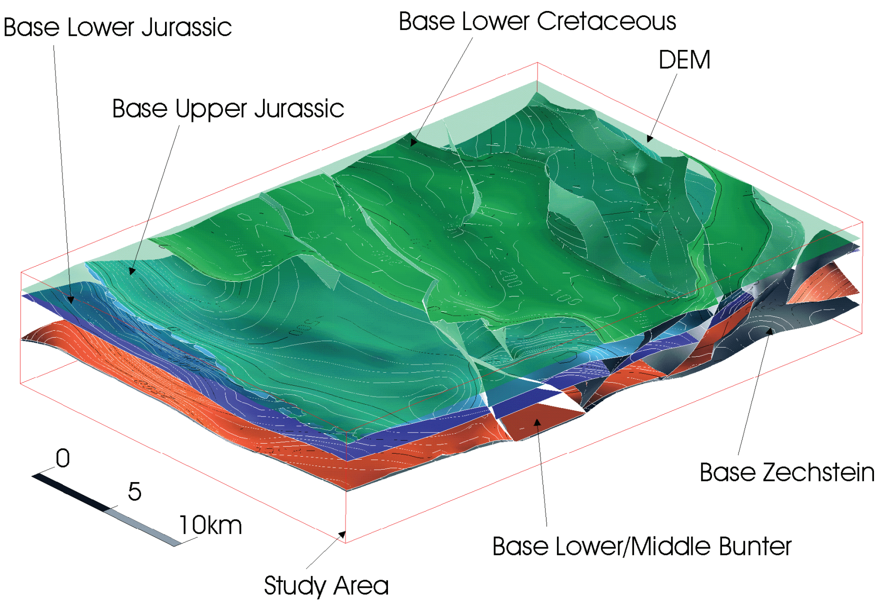

Fig. 4. Geological 3D model of the study area in GOCAD.

This integration was used to support a new approach to geological mapping by a combined use of digital 3D geological and geophysical models. In this approach geological 3D modeling is supported by geophysical modeling of gravity, and vice versa. For example, the geological model from GOCAD (Fig. 4) served as input for subsequent 3D gravity modeling using IGMAS (Fig. 5). This initiates an iterative alternating process, which enables a complex interpretation by integration of different geo-scientific methods, resulting in a better validated geological 3D model. A detailed description of the modeling process can be found at (Schmidt & Götze 1999). The final geological map is constructed by the intersection of the resulting geological 3D model with the Digital Elevation Model (DEM). Moreover, further maps with different contents, like contour maps of stratigraphic boundaries or thickness maps of specific stratigraphic units can be easily derived from the 3D model. The approach is tested with an examination area in Southern Lower Saxony.

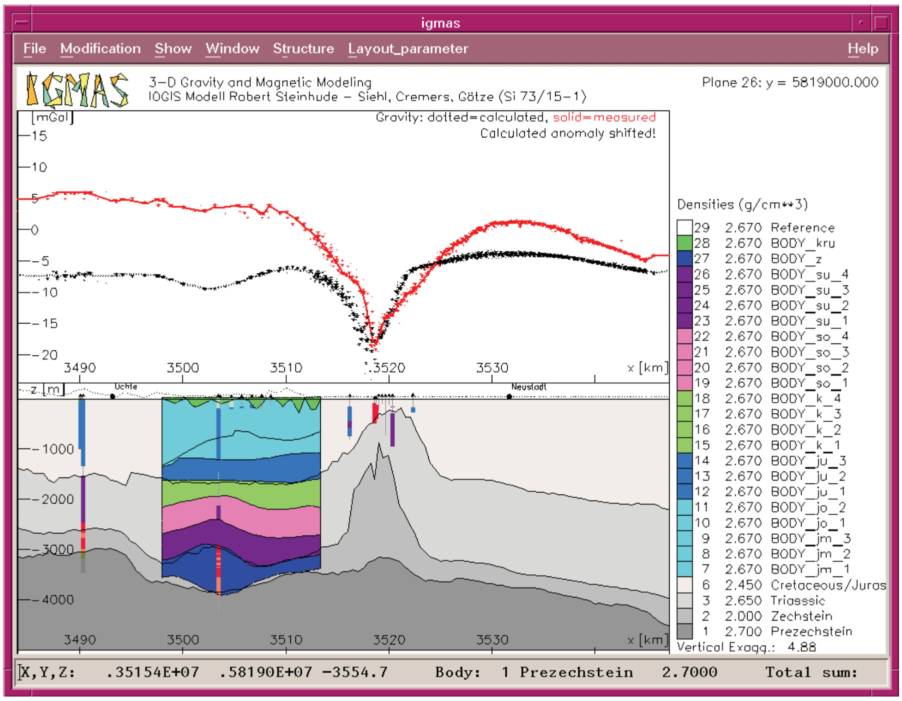

Fig. 5. W-E trending cross section of the IGMAS 3D gravity model, cutting through the study area with the geological 3D model included.

9 Conclusion

We have presented the IOGIS approach to integration of heterogeneous and distributed data sets. An object-oriented modeling technique has been used to describe both, the semantics of geo-scientific meta-data and the needed information of data stored at the different sites. A flexible and extensible system architecture based on software component technology has been developed. It supports integrated access to heterogeneous data on the meta-level as well as on the data source level in order to facilitate both, identification of relevant information and a detailed exploration of data. XML-based data representation within IOGIS allows direct access from web browsers, when XML and XSL support becomes widely available. A prototypic implementation of the client platform and a first version of the server are operable. Currently we are working on the data source framework and the semantic access support for advanced integrated queries.

Finally, the use of Interoperable Open GIS and facilitated exchange of data and models between the applications leads to improvement of geo-scientific results. Geo-Sciences gain distinct benefits by the iterative validation of geo-scientific models using different modeling tools that leads towards a better model's plausibility and validation.

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). Furthermore, we are grateful to M. Assmann, Ch. Michels and U. Radetzki for their implementation and data editing support.

References

Abel D, Kirby PJ, Davis JR (1994): The systems integration problem. In: Intern. Journal of Geographical Information Systems, v3, pp. 291-301.

Amirbekyan V. (1999): Integration of Object Databases with CORBA Environment. Ph.D. Thesis, Stanislaw Staszic University of Mining and Metallurgy, Cracow, Poland.

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.

Breunig, M., Cremers, A.B., Götze, H.-J., Schmidt, S., Seidemann, R., Shumilov, S., Siehl, A. (1999): First Steps Towards an Interoperable GIS - An Example From Southern Lower Saxony. In: Physics and Chemistry of the Earth, Vol. 24, Nr. 3, pp. 179-190.

Bergmann, A., Gärtner, H., Breunig, M., Cremers, A.B., Dikau, R. (1998): Design and First Steps in the Development of OPALIS. In: D. Fritsch, M. Englich, M. Sester (Hrsg.): Proceedings of the ISPRS Commission IV Symposium on ”GIS - Between Visions and Applications”, Stuttgart, Germany, IAPRS, Vol. 32, Part 4.

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

Baralis, E., Paraboschi, S., Teniente, E. (1997): Materialised Views Selection in a Multidimensional Database. In: Proceedings of the 23rd VLDB conference, Athens, Greece, August 1997, pp. 156-165.

Cattell, R.G.G., Barry, D., Bartels, D., Berler, M., Eastman, J., Gamerman, S., Jordan, D., Springer, A., Strickland, H., Wade, D. (eds.) (1997): The Object Database Standard: ODMG 2.0. The Morgan Kaufmann Series in Data Management Systems, 288pp.

Conrad, St. (1997): Föderierte Datenbanksysteme – Konzepte der Datenintegration, Springer, 331 pp.

Gärtner, H., Bergmann, A., Breunig, M., Cremers, A.B. & R. Dikau (2000): OPALIS - An Open Paleoecological Information System. In: GIS No.2/2000.

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

Labio, W.J., Zhuge, Y., Wiener, J.L., Gupta, H., Garcia-Molina, H., Widom, J. (1997): The WHIPS Prototype for Data Warehouse Creation and Maintenance, in: Proceedings of the ACM SIGMOD Conference, Tuscon, Arizona.

OMG (1999): Object Management Group, The Common Object Request Broker: Architecture and Specification, Revision 2.3. June1999.

Rumbaugh, J., Jacobson, I., Booch, G. (1999): The Unified Modeling Language Reference Manual. Addison-Wesley Object Technology Series, Reading, Massachusetts, 576pp.

Schmidt, S., Götze, H.-J. (1999): Integration of Data Constraints and Potential Field Modeling - An Example From Southern Lower Saxony. In: Physics and Chemistry of the Earth, Vol. 24, No. 3, pp. 191-196.

SUN Microsystems Inc. (1997): JavaBeans Specification, http://www.javasoft.com/beans/docs/index.html

SUN (1999): Java 2 Platform – Standard Edition. http://java.sun.com/jdk/

Szyperski, C. (1998): Component Software – Beyond Object-Oriented Programming. Addison-Wesley, New York, 411pp.

W3C (1998): World Wide Web Consortium, Extensible Markup Language (XML) 1.0. www.w3.org/TR/1998/REC-xml-19980210

Wiederhold, G. (1997): Value-added Mediation in Large-Scale Information Systems. In: Robert Meersman and Leo Mark (eds.): Database Application Semantics, Chapman and Hall, pp. 34-56.