Realization of Component-Based GIS Application Framework

Leonid Stoimenov, Aleksandar Stanimirović, Slobodanka Djordjević-Kajan
CG&GIS Lab, Department of Computer Science
Faculty of Electronic Engineering, University of Niš, Serbia and Montenegro
E-mail: leni@elfak.ni.ac.yu, alex@elfak.ni.ac.yu, sdjordjevic@elfak.ni.ac.yu

INTRODUCTION

Increasing number of geodata producers and users have expressed the need for the integration of geodata from distributed information sources and for interoperable GISs (Stoimenov, 2002). The systems that own this data must be capable of interoperation with systems around them, in order to make access to this data to become feasible. The system also must deal with issues unique to geospatial data such as: translating data formats into a uniform transient data structure, consistent coordinate systems, cartographic projections and platform-dependent data representations, and retrieval of associated attributes and metadata (Levinsohn, 2000,Vckovsky, 1998). Integrating geodata from various sources increasingly becomes important because of growing environmental concerns, pressures on governments and businesses to perform more efficiently, and simply because of the existence of a rapidly growing body of useful geodata and geoprocessing tools (Buehler, 1998).

By joining the trend towards interoperation and openness, resource holders gain the ability both to better utilize their own information internally, and to become visible to an increasingly sophisticated user community, no longer satisfied with ringing up, writing to, physically visiting, or working on-line with the proprietary interfaces of a host of providers (Miller, 2000). In this new environment, the interoperable organizations will be visible, usable and customer focused, whilst still maintaining their own unique branding within the portals through which their content is available.

Interoperability of information systems relies on bases of agreement that describe what is shared among information sources. Interoperability means openness in the software industry, because open publication of internal data structures allows GIS users to build applications that integrate software components from different developers. Interoperability also means the ability to exchange data freely between systems, because each system would have knowledge of other systems formats (Buehler, 1998).

The paper is structured as follows. In the second part, we shortly describe related work. The goals of our research activities, described in the third part of this paper, are defining component-based GIS application development framework as part of GeoNis interoperability framework.

RELATED WORK

Research in information systems interoperability is motivated by the ever-increasing heterogeneity of the computer world. Heterogeneity in GIS is not an exception, but the complexity and richness of geographic data and the difficulty of their representation raise specific issues for GIS interoperability. Interoperability means openness in the software industry, because open publication of internal data structures allows GIS users to build applications that integrate software components from different developers. In the past few years the OpenGIS Consortium (OpenGIS Consortium Inc., 1999) (OGC), as a consortium of GIS vendors, agencies, and academic institutions, has emerged as a major force in the trend to openness. Interoperability also means the ability to exchange data freely between systems, because each system would have knowledge of other systems formats. In traditional collaborative engineering, information is managed in an ad hoc manner, with data maintained in application specific files and legacy databases. Data exchange between these systems usually performed by executing scripts and programs that transport data. This approach may work with small projects, but it is inadequate for distributed environments.

Today, the only way to integrate huge amount of available data is to build specialized applications based on mediators.

Existing GIS applications have to be modified in order to implement interoperability. Possible solution is implementation of wrappers or data translators that provide “interface” for interoperable environment. During development of new GIS applications, parts of the system that will provide interoperability must be considered.

Modern GIS applications have to deal with great number of heterogeneous data sources. Process of designing that kind of applications can become very complicated. It is very useful, in order to avoid this complication, to realize data access in separate component. In this way, application becomes completely independent from underlying data access technique. That kind of components will provide means for development of new applications, capable to interchange data with systems from distributed environment.

Component-oriented programming is the predominant software development methodology today (Stojanovic, 2002, Szyperski, 1998). Components can be considered as stand-alone service providers. When a system needs some service, it calls on a component to provide that service. In that case system doesn’t take care about where component is executing or about programming language that was used for component development. Component could be considered as a completely independent executable entity. Components publish their interfaces and all interactions are through that interfaces. Components may exist at different level of abstraction, from a simple library subroutine to an entire application.

There are many reasons why component-based software development is desirable. For example, components have often been used in other systems and have been extensively tested. This methodology also allows much greater reusability, extensibility, and maintainability. Component-based systems are more robust and highly scalable. With the correct choice of components, experts in specific application domains may be able to build whole new application systems adding very small amount of code as glue for existing components.

As a solution for this problem OpenGIS Consortium has proposed two standards (OpenGIS Consortium Inc., 1999): OpenGIS Abstract Specification for Features and Open GIS Simple Feature Specification for OLE/COM. OpenGIS Simple Feature specification provides unified method for accessing data spatial data stored in both databases and proprietary formats. This technology is based on Microsoft OLE DB standards. This standard provides data in unified table-oriented manner to any system or application.

Universal Data Access (UDA) is the Microsoft strategy for providing access to all types of data across information system (Rauch, 1997). Microsoft UDA provides access to a variety of information sources including relational and non-relational data (e-mail and file system stores, text, graphical, geographical data and more). The Universal Data Access provides component architecture. This kind of architecture allows components to implement only required set of functionalities over data sources. UDA also assumes existence of service components that implements additional functionalities on the top of less capable components.

The Microsoft UDA strategy is based on OLE DB. Microsoft defines OLE DB as a specification for a set of data access interfaces (Microsoft Press, 1999). This set of interfaces expose data from a variety of sources using OLE Component Object Model (COM). These interfaces provide applications with uniform access to data stored in diverse information sources.

GEONIS FRAMEWORK

The total number of geodata providers in local community environment is indeterminable and unlimited. This implies the need for a flexible approach that can deal with the existing and the future geodata providers in interoperable systems. GeoNis is framework for interoperability of GIS applications that have to provide infrastructure for data interchange in the local community environment (Stoimenov, 2000). Data sources are local services and offices that own geodata in some format.

GeoNis solution to the problem of semantic heterogeneity is to formally specify the meaning of the terminology of each GIC (GI community, i.e. local service or office) using local ontology and to define a translation between each GIC terminologies (local ontologies) and an intermediate terminology (in top-level ontology). A semantic translation in GeoNis is developed for a particular domain, in our case for GIS applications in local community services, which deal with network data structures (local Telecom, water and soil pipe services, power supply services, and some local government services) (Stoimenov, 2003).

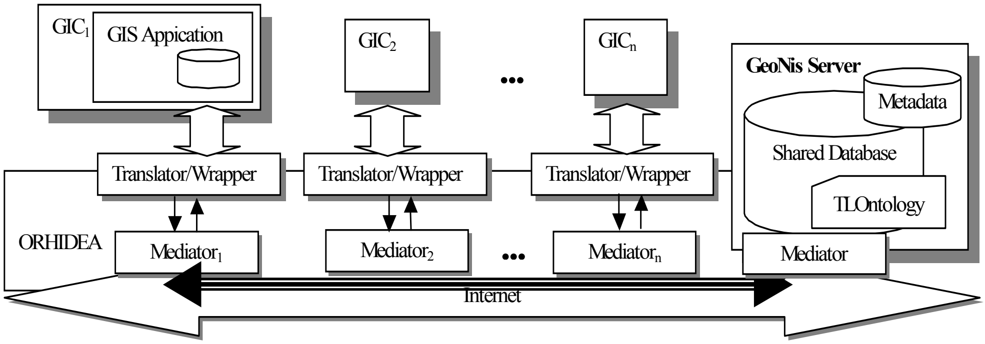

Figure 1: Architecture of GeoNis framework.

The basic architecture of GeoNis framework is shown on Fig. 1. Each GIC contain GIS application and corresponding (spatial) database. GeoNis is based on ORHIDEA (Stoimenov, 2002) data integration platform. ORHIDEA platform has been developed in order to perform intelligent integration of information from multiple heterogeneous GIS (spatial and geographic), and non-spatial (thematic) data sources.

Each type of information source requires a wrapper/translator that translates information flow between information source and GeoNis system. In each node of GeoNis framework there exists GIS application and corresponding (spatial and non-spatial) databases. Data in local databases are accessible according to user privileges. Requests for specific data set are forward through local mediators. These applications may be either wrapped legacy applications or newly developed applications based on Ginis application development framework.

The Ginis is a component-based, generic information integration framework for integrating existing local information sources. Information sources use the same ontology and there is not need for semantic translation in Ginis.

In addition to domain oriented GIS applications, there is one common GIS server that maintains all shared/common geographic data. These data are public available and could be used by GIS clients in every GIC or citizens through the available public services on user demand. Access to data in local spatial databases depends on user privileges. Requests for specific data set are forward through local mediators or GeoNis server. GeoNis server also contains information about registered GIC and their access rights. Every new GIC who wants to participate in exchanging data must register on GeoNis server in order to allow access to his public available data and local ontology. After that, registered GIC have access to all available data from other public GIC databases (with possible given rights for access), and access to shared data on GeoNis server.

GINIS – COMPONENT-BASED GIS APPLICATION FRAMEWORK

Work in Ginis project is based on our previous work in realization of object-oriented GIS application framework (Stoimenov, 1999,), while the underlying database is actually stored and maintained by a

RDBMS. The Ginis is an object-oriented GIS framework that provides transparent access to variety of data sources. Ginis is realized using component-based technology. The primary difference between the Ginis and traditional GISs is that the Ginis can be dynamically connected to data sources, and may dynamically change the user interface according to user privileges.

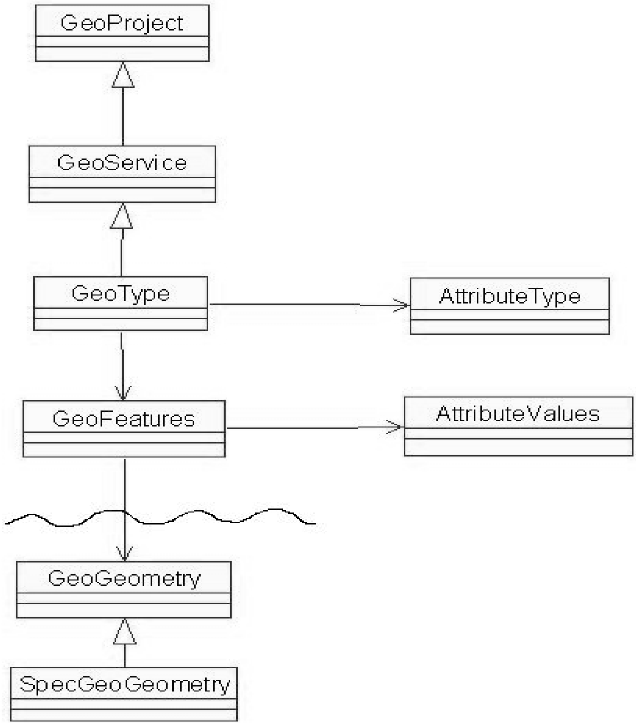

Figure 2: Subset of Ginis classes.

Our object-oriented GIS framework allows encapsulation of geo-spatial entities so that all of its geometry, data, and behaviors are contained in a single object. Ginis uses simple class hierarchy (subset of this hierarchy is shown on Fig. 2), for each project or for each user with defined privileges. This hierarchy is only for memory representation of project classes i.e. for instantiation of a project.

Information about objects on a map would be stored in various files and databases. The data could be retrieved and used for presentation, or modeling and analysis, by selecting one or more objects displayed on the map. Ginis application is designed in the way that a data representation is completely separated from data storage. Architecture of Ginis application is shown in Fig. 3 and here we can distinguish two components: GeoContent component and DataAccess component. Also, there are two service-based modules: Client Viewer and SecurityManager.

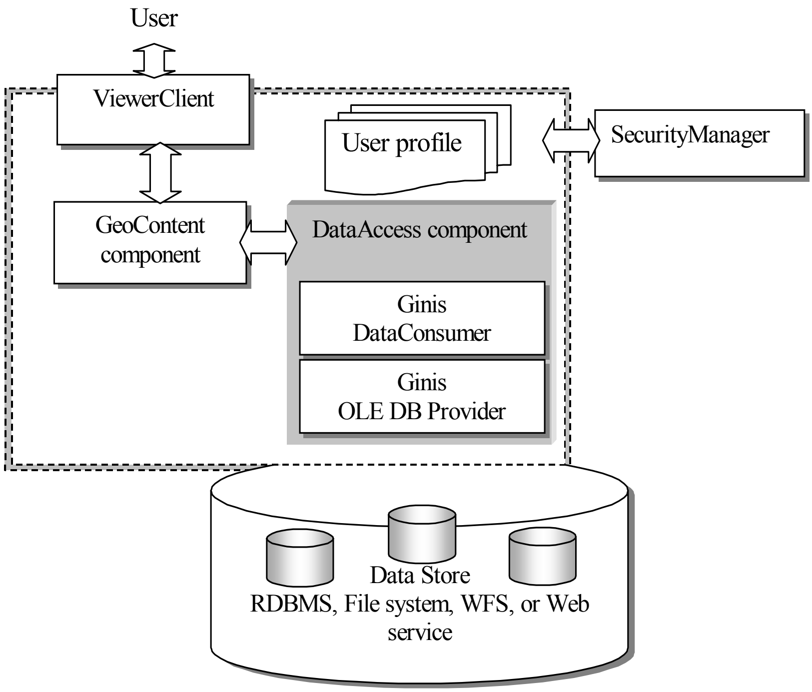

Figure 3: Architecture of Ginis application.

GeoContent component is responsible for representation of spatial data. User interface of GIS application (ViewerClient) directly communicate with this component in order perform operations requested by the user. This component implements facilities for spatial data manipulation, facilities for geo-analysis, etc. A rich layer hierarchy allows arbitrary grouping of objects based on any relationship as well as the traditional GIS concept of layering. GeoContent component perform data operations only through DataAccess component. In this way GeoContent component is completely independent from various types of data sources.

DataAccess component is responsible for data operations. This component consists of two modules: Ginis OLE DB DataProvider and Ginis DataConsumer. The DataConsumer is a layer between GeoContent component and Ginis OLE DB Provider. This way GeoContent component becomes not only independent from different data sources but also independent from underlying technology for accessing this various data sources. Ginis OLE DB Provider provides access to diverse data sources. In order to achieve this functionality this component use existing commercial OLE DB providers. This component also provides to other components access to file system is responsible for providing data consumers with GIS specific functionality.

The basic architecture of Ginis OLE DB provider is shown in Fig. 4. Ginis OLE DB provider consists of three components:

1. Data Manager Component,
2. OLE DB Component and
3. GIS DB Component.

Data Manager Component provides access to diverse data sources to OLE DB and GIS DB components. In order to achieve this functionality this component use existing commercial OLE DB providers. This component also provides to other components access to file system.

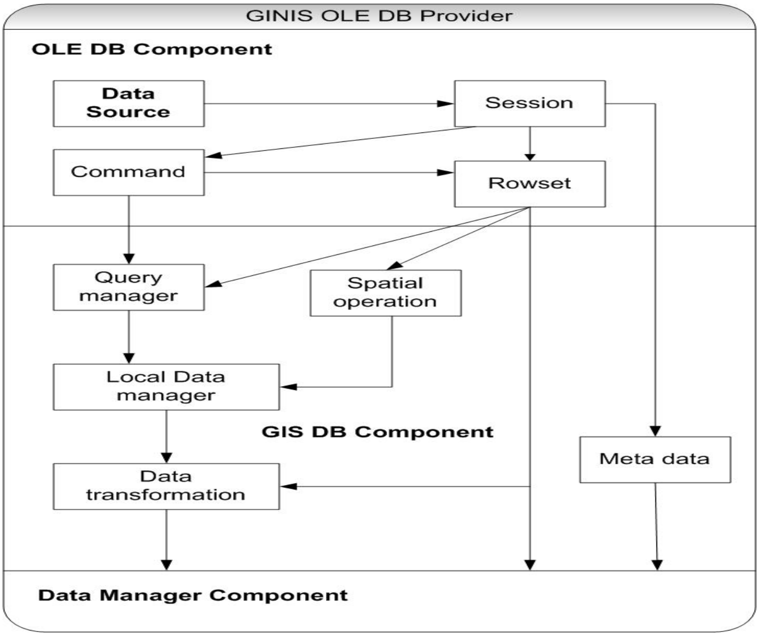

Figure 4: Ginis OLE DB provider basic architecture.

OLE DB Component exposes standard interfaces to data consumers according to Microsoft (Rauch, 1997). This component implements Data Source, Session, Command and Rowset objects. Data Source object initialize and sets up connection to appropriate database. Session object manage database concern metadata and creates Command object for data manipulation. Command object allows invoking execution of data-manipulation and data-creation commands. Rowset objects manage result features that are transferred form database as result of Command object execution.

GIS DB Component is responsible for providing data consumers with GIS specific functionality. In order to do this GIS DB component must extend or implement new interfaces besides interfaces defined by Microsoft specification:

- GIS Meta data module – Session object provides all meta-information about data source. Original implementation of OLE DB Session object must be extended in order to provide additional GIS specific meta-information. GIS Meta data module is responsible for extending Session object with this additional functionality.
- Data transformation module – This module transforms data obtained from Data Manager Component into WBK typed geometry data according to OGC specification.
- Local data manager module – This module is responsible for keeping local copies of the data obtained from data sources. Local copies of data are necessary in order to perform some additional data transformations.

- Query manager module – This module is responsible to provide facilities for query execution. Query manager module process all query requests that came from Command object and decides where and how this queries will be executed. This module must provide facilities for dealing with distributed queries and facilities for query execution against local data. At the moment only transformation part of this module is implemented. This part is responsible for transforming queries, expressed in different query languages (XQuery for example), into SQL queries.
- Spatial operations module – This module will be responsible for implementation of different spatial operators and spatial filters. It is not implemented yet.

GIS DB Component is responsible for providing data consumers with GIS specific functionality. In order to do this GIS DB component must extend or implement new interfaces besides interfaces defined by Microsoft specification.

The ViewerClient is a Ginis instance and it can be run as a stand-alone or Web based client application. ViewerClient is a user interface component that renders, or possibly simply displays, graphics that come from GeoContent server components. Functionality of this component includes layer control capability, fusion of multiple layers, interface for locating datasets of interest, etc. ViewerClient allows changing attribute values, changing and deleting any geo-object. ViewerClient component has Legend object class, which maintain the layer hierarchy or layer drawing order.

Security Manager represents user profile manager. Security manager is implemented as service-based component and may be used by different user applications. All user applications (or Ginis instances) from a GIC are connected to the same Security manager (part of GeoNis Server). The Security manager uses attributes and associated meta-data to dynamically build a user profile to customize GIS application. This profile is send to user application as an XML coded file. According to this file, Ginis customizes application interface i.e. adjust menu items and dialogs according to user privileges.

CONCLUSION

Being seen to "be interoperable" is becoming increasingly important to a wide range of organizations, including central and local government. A truly interoperable organization is able to maximize the value and reuse potential of information under its control. It is also able to exchange this information effectively with other equally interoperable bodies, allowing new knowledge to be generated from the identification of relationships between previously unrelated sets of data.

Requests for interoperable applications become very important to great number of organizations that utilize GIS information (including public services, local communities, government, etc.). Fully interoperable organization is capable to maximize value and reusability of geoinformation that owns. Such organization is also capable to interchange information of interest with other interoperable organizations that exists in the same environment, increasing, that way, quality and range of knowledge generated by identifying relations between existing and new data.

In order to achieve interoperable organization, platform that will provide means for data interchange between various geographic information systems and applications must be implemented. Component based GIS environments provide means for development of GIS applications that are capable to support interoperability.

To provide integrated access to various distributed geo-information sources, we have developed components as an extension to existing GIS application called Ginis. This application development framework is part of GeoNis framework for interoperability of GIS applications. The Ginis has an object-oriented architecture that is based on modern software component technologies. This paper describes platform and component based application that provide access to various geo-information sources. Proposed component architecture completely separates representation of geodata from data stores. This kind of architecture can be reused in different GIS applications.

ACKNOWLEDGEMENTS

This research was partially supported by the project “Geographic Information System for Local Authorities based on Internet/WWW Technologies”, funded by Ministry of Science, Technology and Development, Republic of Serbia, Contract No. IT.1.23.0249A.

BIBLIOGRAPHY

Buehler R., and McKee L., 1998, The Open GIS Guide, Third Edition, OpenGIS Consortium, Inc, http://www.OpenGIS.org/techno/specs.htm

Levinsohn A., 2000, «Geospatial Interoperability: The Holy Grail of GIS», GeoEurope, available online: http://www.geoplace.com/gw/2000/1000/1000data.asp

Microsoft Press, Microsoft OLE DB 2.0 Programmer’s Reference and Data Access SDK, 1998

Miller, P., 2000, "Interoperability. What is it and Why should I want it?", Ariadne Issue 24, Publication date : 21-Jun-2000, http://www.ariadne.ac.uk /issue24/interoperability/intro.html

OpenGIS Consortium Inc, OpenGIS Simple Features Specification For OLE/COM, 1999, http://www.opengis.org

Stoimenov L., Mitrovic A., Djordjevic-Kajan S., Mitrovic D., Bridging objects and relations: a mediator for an OO front-end to RDBMSs, Information and Software Technology, Elsevier, 1999, Vol 41, No. 2, pp. 59-68.

Stoimenov L., Djordjević-Kajan S., Stojanović D., Integration of GIS Data Sources over the Internet Using Mediator and Wrapper Technology, MELECON 2000, Cyprus, 2000.

Stoimenov L., and Đorđević-Kajan S., 2002. Framework for semantic GIS interoperability, FACTA Universitatis (Niš), Series: Mathematic and Informatics, 17(2002), pp.107-125.

Stoimenov L., and Đorđević-Kajan S., 2003, Realization of GIS semantic interoperability in local community environment. In: Proceedings 6th AGILE Conference, Lion, France, pp.73-80.

Stojanovic Z., and Dahanayake A.N.W., 2002, "A New Approach to Components", accepted for the 2002 Information Resources Management Association (IRMA) International Conference, to be held in Seattle Washington, USA, May 19-22, 2002.

Szyperski, C., 1998, "Component Software: Beyond Object-Oriented Programming", ACM Press, Addison-Wesley.

Rauch S., 1997, Manage Data from Myriad Sources with the Universal Data Access Interfaces, Microsoft System Journal, September 1997.

Vckovsky A. (ed.), 1998, International Journal of Geographic Information Science - Special Issue: Interoperability in GIS, Vol 12, No 4, 1998.