INTEROPERABLE 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

SUMMARY

In this paper we present research in Geographic Information Systems (GIS) interoperability. Also, this paper describes interoperability framework called GeoNis. GeoNis uses new technologies, proposed in this paper, to perform integration task between GIS applications and legacy data sources over the Internet. Our approach provides integration of distributed GIS data sources and legacy information systems in local community environment. The proposed framework uses the technology based on mediators, to allow communications between GIS applications over the Internet/Intranet. The problem of semantic heterogeneity will be resolved by concept of mediation and ontology.

To provide integrated access to various distributed geo-information sources, we have developed components as an extension to existing GIS application called Ginis. Component-based architecture of Ginis is also presented in this paper. We have developed DataConsumer module that encapsulates physical data access details from the rest of application. In this way we have completely separated GIS application from details of data access.

Also we have described our implementation of OpenGIS standards for uniform access to heterogeneous and distributed data sources. These standards are based on Microsoft Universal Data Access specification and OLE DB technology. Existing GIS application has been extended with new components for data access. Crucial part of this implementation is Ginis OLE DB Data Provider that is responsible for providing spatial data. The basic architecture of Ginis OLE DB Data Provider is also shown in this paper.

Keywords: *nteroperability framework, component-based development, mediation, geographic information systems, Ginis, LE DB*

1. INTRODUCTION

Geographic information systems (GIS) are computerized systems for managing data about spatially referenced objects. GIS data are typically used by various groups of users with different views and needs. Also, GIS applications are often built on different software platforms and execute on different hardware platforms. Nowadays, there is a strong trend of information systems integration in chain of systems among public information structures such as Internet. Another trend in GIS is publishing maps for World Wide Web community and development of web-based GIS applications.

Research in information systems interoperability is motivated by the ever-increasing heterogeneity of the computer world. Interoperability means openness in the software industry, because open publication of internal data structures allows users to build applications that integrate software components from different developers [15].

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. Popularity of GIS in government and municipality institutions induce increasing amount of available information [20]. In local community environment (city services, local offices, local telecom, public utilities, water and power supply services, etc) different information systems deal with huge amount of available information, where significant amount of this data is geo-referenced. Information that exists in diverse data sources may be useful for many other GIS applications. But, information communities find it difficult to locate and retrieve data from other sources, in reliable and acceptable form. Each of these user groups has a different view of the world and available information is always distributed and mostly heterogeneous.

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. These applications also must deal with issues unique to geospatial data, including 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 [7][26].

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 [9]. 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.

Component-oriented methodology is predominant programming methodology today. It allows reusability, extensibility, and maintainability. 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 is part of GeoNis interoperability platform. In this paper we present GeoNis and component-based architecture of Ginis.

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. In fourth part of this paper we described extension of component GIS application with components for data access and our proposal for basic architecture of Ginis OLE DB Data Provider.

2. RELATED WORK

The need to share geographic information is well documented [26]. Recent reviews of GIS interoperability and integration efforts can be found in [1] [6]. Today, research on interoperability solutions is the way to migrate away from the monolithic systems that dominate the GIS market [15]. In the monolithic GIS system, all layers are tightly coupled and not open to other systems, except through the very elemental means of exchanging data with identical monolithic systems or translating data from other systems [2].

The realization of interoperable GI systems is weighty process, as a consequence of two main system characteristic - distributed data sources and their heterogeneity [5]. Information systems heterogeneity may be considered as structural (schematic heterogeneity), semantic (data heterogeneity), and syntactic heterogeneity (database heterogeneity). Syntactic heterogeneity means that various database systems use different query languages (SQL, OQL, etc). Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content of an information item and its meaning. Semantic conflicts among information systems occur whenever information systems do not use the same interpretation of the information.

Mediator-based system is important for spatial data interoperability architecture [20]. Mediator-based systems are constructed from a large number of relatively autonomous sources of data and services, communicating with each other over a standard protocol and enabling “on-demand” information integration [29]. Structural and syntactic heterogeneity may be solved by mediation. The 3-level architecture of mediator-based systems is constructed from an application layer, and large number of information sources (heterogeneous data sources with wrappers), communicating with each other over a standard protocol. Nowadays, mediation concept is a part of the ARPA I3 (Intelligent Information Integration) reference architecture. The I3 reference architecture should be seen as a vision of how vast amount of heterogeneous information can be incrementally pulled into a gigantic, reusable library of information resources.

Component-oriented programming is a programming paradigm which has as its goal to develop software by means of software components, believing that this will increase speed of development and reliability of the resulting computer systems and computer programs [24] [25]. There are many reasons why build software from components 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. Possibly with the correct choice of components, experts in specific application domains may be able to build application systems.

Component-based development platforms and technologies are now de facto standards for implementation and deployment of complex enterprise distributed systems. Today, there are four predominant, competing, technologies for component-oriented software development [4]:

1. CORBA (Common Object Request Broker Architecture) – developed by OMG (Object Management Group)
2. COM/DCOM (Component Object Model/Distributed Component Object Model) – developed by Microsoft
3. Enterprise Java Beans (EJB) – developed by Sun Microsystems.
4. NET – developed by Microsoft.

These technologies define bases for communication and cooperation between software components from different vendor. At the same time these specifications provides methods for building software from existing components. CORBA, COM/DCOM and .NET components can be developed using different programming languages and JavaBeans can be developed only using Java language. COM/DCOM and .NET architecture are specific for Windows platform and CORBA and JavaBeans components can be executed on any platform.

XML (eXtensible Markup Language) was originally designed as a solution for adding extensions to HTML, but for past several years is rapidly becoming the technology integrating heterogeneous component-based systems [28][3]. XML define minimal level of standardization for data and message interchange that can ensure communications between components. The core XML specification is extremely simple, as it only lays down the syntactic ground rules for forming valid XML messages. While the World Wide Web Consortium (W3C) is rapidly layering additional standards on top of XML (for example, XLink and XML Schemas), the base-XML syntax has been fairly stable. The base XML syntax has proven to be quite flexible and adaptable to many applications, and despite its hierarchical nature, XML lends itself reasonably well to non-hierarchical data types.

Web services are the fundamental building blocks in the move to distributed computing on the Internet [27]. Open standards and the focus on communication and collaboration among applications, have created an environment where Web services are becoming the platform for application integration. Applications are constructed using multiple Web services from various sources that work together regardless of where they reside or how they were implemented. Web services use XML standards to provide mechanism for communication and cooperation between components written in different languages and executing on different platforms.

One important initiative to achieve GIS interoperability is the OpenGIS Consortium (OGC) [10]. This is an association looking to define a set of requirements, standards, and specifications that will support GIS interoperability. The objective is technology that will enable an application developer to use any geodata and any geo-processing function or process available on 'the net' within a single environment and a single workflow [2]. But, data standardization is not the whole solution. The interoperability problem would go away if every system always uses the same data model to represent the same information (identical names, structure, and representations).

According to OGCs OpenGIS Abstract Specification for Features and OpenGIS Simple Feature Specification access to spatial data should be seen in the context of database problems [10]. Spatial geometry should be seen as one of the object attributes and feature collections could be easily represented as tables in relational database. Several feature collections (tables) also form one relational database. This point of view is supported by current trends in database systems. Modern relational database management systems introduce new technologies for spatial data storage. OpenGIS Consortium has proposed OpenGIS Simple Features Specification For SQL [12] to standardize storing geo-spatial data in relational databases. Unfortunately there exists large base of spatial data stored in proprietary formats (mostly binary data in various GIS formats). Chances that this data will be converted to some standard database format are very small.

For this reason OGC proposed two specifications: OpenGIS Simple Feature Specification For OLE/COM and OpenGIS Simple Feature Specification For CORBA [10]. These specifications provide architecture for developing components that expose functionality, for accessing and manipulating spatial data stored both in databases and proprietary formats.

OpenGIS Simple Feature Specification For OLE/COM specification is based on Microsoft Universal Data Access (UDA) architecture [11]. UDA is the Microsoft strategy for providing access to all types of data across information system [13]. 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 [8]. 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 in table-oriented manner.

3. GEONIS FRAMEWORK

Our primary goal with GeoNis project was to design a platform and methodology that could serve for information exchange over the Internet between distributed and Web-based GIS sources in local community [18]. However, the Web's browsing paradigm does not readily support retrieving and integrating data from multiple sites. The only way to integrate the huge amount of available data is to build specialized platforms.

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 [21]. Data sources are local services and offices that own geodata in some format. 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) [19]. Each type of information source requires a wrapper/translator that translates information flow between information source and GeoNis system. GeoNis solution to the problem of semantic heterogeneity is to formally specify the meaning of the terminology of each GIC using local ontology and to define a translation between each GIC terminologies (local ontologies) and an intermediate terminology (in top-level ontology). Architecture of GeoNis interoperability framework is described in [19].

A standard model for spatial data is the first step to approach the solution for schematic and syntactic heterogeneity. The Open GIS Consortium specification [2] aims to solve the problem of heterogeneity at the spatial data modeling level. Because of that, GeoNis uses OpenGIS standard as common data model to represent geodata on mediator level. Data models of local information sources are translated in common model using wrappers.

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.

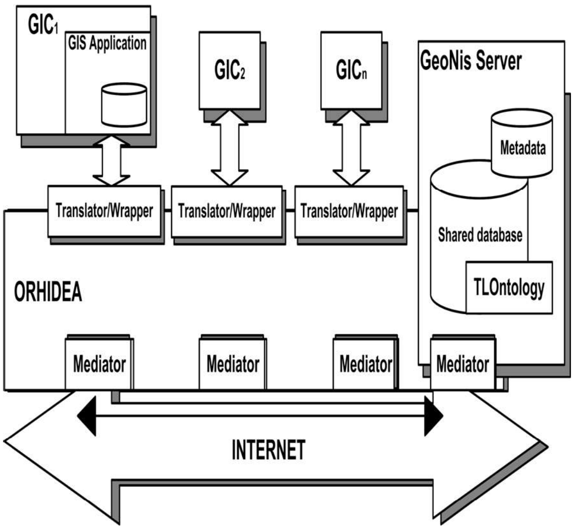

Fig. 1 GeoNis integration platform

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.

4. GINIS – COMPONENT-BASED GIS APPLICATION FRAMEWORK

Work in Ginis project is based on our previous work in realization of object-oriented GIS application framework [18][19][22], 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 can be dynamically connected to data sources, and may dynamically change the user interface according to user privileges.

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. 1), 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.

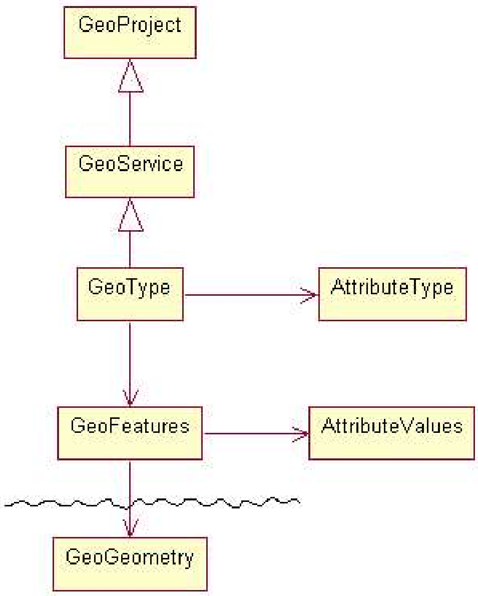

Fig. 2 Subset of Ginis Classes

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 Figure 2 and here we can distinguish two components: GeoContent component and DataAccess component. Also, there are two service-based modules: Client Viewer and SecurityManager.

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. 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.

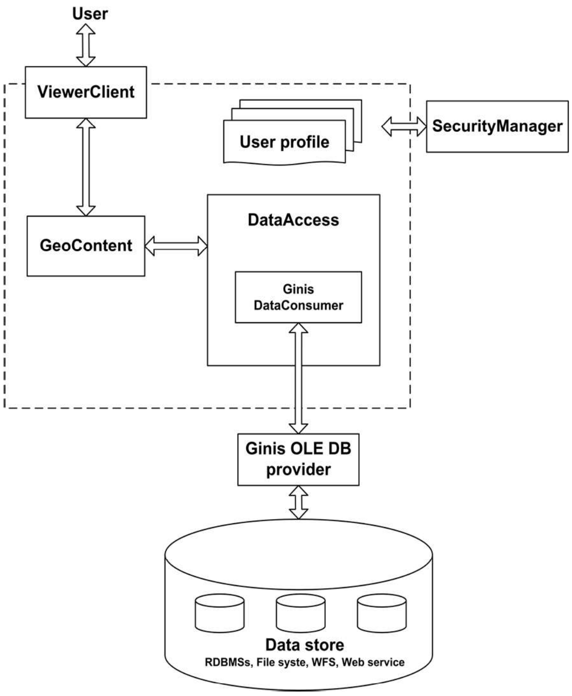

Fig. 3 Architecture of Ginis

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, among others, contains Ginis DataConsumer module. The DataConsumer is a layer between GeoContent component and Ginis OLE DB Provider. This way GeoContent component is not only independent from different data sources but also independent from underlying technology for accessing this various data sources.

4.1 DataConsumer module

Implementation of DataConsumer module is based on several different design patterns: Iterator, DataAccessor (Adaptor), Singleton and Factory [16]. Iterator pattern allows application to use unique interface for traversing and accessing data obtained from various data sources. At the same time Iterator pattern hides from application mechanics involved in retrieving data from various data sources. DataAccessor is a design pattern that adapts interfaces of different data providers to the interface required by Iterator. Factory pattern is responsible for creating appropriate implementation of DataAccessor interface according to nature of data source that is used by application.

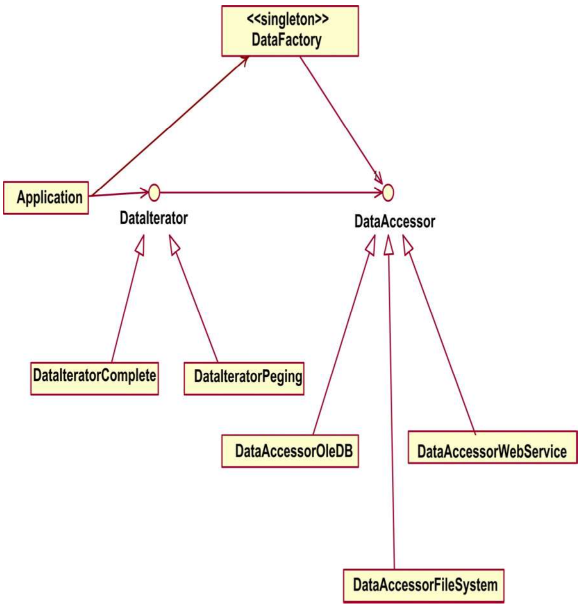

Fig. 4 Implementation of Ginis DataConsumer

Fig. 4 illustrates some of the classes that are part of Ginis DataConsumer implementation:

- DataIterator – defines interface for accessing and traversing data.
- DataIteratorComplete, DataIteratorPaging – implementations of DataIterator interface that use different strategy for traversing data.
- DataAccessor – defines interface for retrieving data from different data sources.
- Implementations of DataAccessor interface for different data sources:
  - DataAccessorOleDB,
  - DataAccessorWebService,
  - DataAccessorFileService
- DataFactory – Singleton, which is responsible for creating appropriate implementation of DataAccessor interface according to data source that, is used by application.

4.2 Ginis OLE DB Provider

Ginis OLE DB Provider is component developed according to OpenGIS Simple Features For OLE/COM specification. This component provides access to diverse data sources using existing commercial OLE DB providers and other providers that are native for data source. It is also responsible for providing data consumers with GIS specific functionality [23]. Basic architecture of this component is shown on Figure 5 [17].

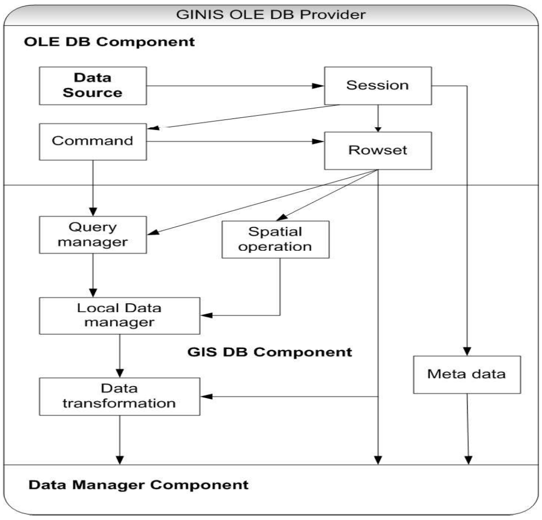

Fig. 5 Architecture of Ginis OLE DB Provider

Data Manager module 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.

OLE DB module exposes standard interfaces to data consumers according to Microsoft. 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 module 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 – 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 – This module transforms data obtained from Data Manager Component into WKB (Well Known Binary) or WKT (Well Known Text) typed geometry data according to OGC specification.
- Local data manager– 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 – 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.

5. 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.

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.

At the same time we have proposed architecture of Ginis application development framework. This framework. This framework and component-based development offers several advantages:

- Rapid development of Ginis applications using existing components.
- New GIC nodes can easily be added to existing GeoNis infrastructure.
- Ginis applications are independent from heterogeneous and distributed data sources.

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.

REFERENCES

[1] Abel, D.J, Ooi, B.C., Tan, K.L., Tan, S.H., 1998. Towards integrated geographical information processing in International Journal of Geographical Information Science 12 (4), 353-371.

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

[3] Box D., Lessons from the Component Wars: An XML Manifesto, Microsoft Presss, MSDN Library, XML General Technical Articles, September 1999

[4] The Component-base Development Headquar-ters, http://www.cbd-hq.com

[5] Genesereth M., Keller, A.M., Duschka, O.M., 1997. Infomaster: an information integration system, In: Proceedings ACM SIGMOD Conference, pp.539-542.

[6] Laurini R., 1998. Spatial multi-database topological continuity and indexing: a step towards seamless GIS data interoperability in International Journal of Geographic Information Science 12 (4), 373-402.

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

[8] Microsoft Press, Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK, 1998

[9] 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

[10] OpenGIS Consortium Inc., http://www.OpenGIS.org

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

[12] OpenGIS Consortium Inc, OpenGIS Simple Features Specification For SQL, 1999, http://www.opengis.org

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

[14] Shelton, Robert E. "The Distributed Enterprise (Shared, Reusable Business Models the Next Step in Distributed Object Computing)." Distributed Computing Monitor 8, 10 (October 1993): 1

[15] Sondheim M., Gardels K., Buehler K., 1999. GIS Interoperability. In: Longley P., Goodchild M., Maguire D., Rhind D. (Eds.), Geographical Information Systems 1 Principles and Technical Issues, John Wiley & Sons, New York, pp. 347-358.

[16] Stanimirovic A., Đorđević-Kajan S., Stoimenov L., Applying Data Accessor design pattern for GIS OLE DB provider implementation, YUINFO 2004, Kopaonik, Yugoslavia, 2004, CD Edition

[17] Stanimirovic A., Đorđević-Kajan S., Stoimenov L., Ginis OLE DB data provider, ETRAN 2004, Čačak, Yugoslavia, 2004

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

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

[20] Stoimenov L., Djordjevic-Kajan S., Milosavljevic A., 2002., Realization of Infrastructure for GIS Interoperability, In: Proceedings YUINFO '02, Kopaonik, Yugoslavia, CD Edition.

[21] Stoimenov L., Djordjevic-Kajan S., Stojanovic D., 2000. Integration of GIS Data Sources over the Internet Using Mediator and Wrapper Technology. In: Proceedings MELECON 2000, 10th Mediterranean Electrotechnical Confe-rence, Cyprus, Vol.1, pp.334-336.

[22] Stoimenov L., Stanimirovic A., Đorđević-Kajan S., Development of component-based GIS interoperability framework, YUINFO 2004, Kopaonik, Yugoslavia, 2004, CD Edition

[23] Stoimenov L., Stanimirović A., Đorđević-Kajan S., Realization of Component-Based GIS Application Framework, Conference Proceedings, F. Toppen, P. Prastacos [Editors], 7th Agile Conference on Geographic Information Science, ISBN 9-60524-176-5, Heraklion, Crete, 29 April-1 May, 2004, Crete University Press, pp. 113-120

[24] Stojanovic Z., Dahanayake A.N.W., 2002, "A New Approach to Components", 2002 Information Resources Management Association (IRMA) International Conference, Seattle Washington, USA, May 19-22, 2002.

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

[26] Vckovsky A., 1998, International Journal of Geographic Information Science - Special Issue: Interoperability in GIS, Vol 12, No 4, 1998.

[27] Wolter R., XML Web Services Basics, Microsoft Press, MSDN Library, Web Services Technical Article, December 2001

[28] World Wide Web consortium (W3C), http://www.w3c.org

[29] Wiederhold, G. 1998. Weaving data into information in Database Programming & Design 11 (9), pp.22-29.

BIOGRAPHY

Leonid Stoimenov received the BSc, MSc and PhD degrees in computer science at the University of Niš, Serbia. Hi is currently an associate professor in the Department of Artificial Intelligence at Faculty of Electronic Engineering at this University. His research interests in computer science include GIS, databases, mediators, ontologies and semantic interoperability. He is a member of IEEE and representative in AGILE association of GIS laboratories in Europe.

Aleksandar Stanimirović received the BSc degree in computer science at the University of Niš, Serbia. His research for MSc degree is concerned with spatial data modeling as well as design and development of methodologies and tools for managing spatial data. Since 2000 he is working as a teaching assistant with the Department of Computer Science. He is a student member of IEEE.

Slobodanka Đorđević-Kajan received the BSc, MSc and PhD degrees in computer science at the University of Niš, Serbia. She is currently a full professor in the Department of Databases and Software Engineering and head of Computer Graphic and GIS Lab. Hers research interests in computer science include GIS, databases in general, and software engineering. He is a member of IEEE.