Data Access Layer Generation for Interoperable GIS Environments

Aleksandar Stanimirović, Miloš Bogdanović, Leonid Stoimenov
Faculty of Electronic Engineering, University of Niš {aleksandar.stanimirovic@elfak.ni.ac.rs, milos.bogdanovic@elfak.ni.ac.rs, leonid.stoimenov@elfak.ni.ac.rs}

INTRODUCTION

Nowadays, information systems (IS) use the Internet/Intranet as their natural operating environment. Internet/Intranet environment has introduced additional opportunities in the development and operation of information systems and sources of information they use. Information systems take advantage of multiple heterogeneous sources of information. Using information from different sources includes the ability of these applications to integrate information. Effective information integration from heterogeneous sources relies upon semantics of data that is to be integrated.

In order to enable information integration, it is necessary to make information both machine readable and machine understandable. Ontology and semantic languages give us an opportunity to define a meaning of the information and machine-processable semantics of information sources. Ontologies are used by different ISs that participate in communication in order to solve the meaning of the available data. Ontology development and usage is also present in the area of a Geo-Information System (GIS). Geo-information domain has characteristics that further complicate the semantic description of geospatial information necessary to achieve GI systems interoperability.

Geospatial data semantics deals with representations of geographical world as interpreted by human users. Representation and reasoning on the meaning of geospatial data are critical for the development of interoperable geospatial data and software, geographical information retrieval, and automated spatial reasoning. This unavoidably includes development of spatial ontologies. Well designed spatial ontologies could supply both stand-alone applications and search engines with localization information and thus refine queries and enrich the retrieved information.

However, as a result of complex, heterogeneous, multimodal and multimedia representations of spatial data, it is very difficult to capture and maintain semantic knowledge of geographical data. At the moment, geographical information systems either impose a simple semantic structure or do not address the issue of semantics at all, but simply offer textual metadata description, leaving the burden of meaning construction to the user. Such solutions are inadequate in the current stage of geographical information technologies where massive exchange of data from heterogeneous sources must be supported.

Ontology usage intensification in the IS development could be achieved if we treat ontologies not only as a means of semantic mediation of the existing ISs, but as a starting model for the IS development process. This approach, used for information integration, demands ontology-to-data source mapping. The process of ontology-to-data source mapping is the first step in making IS data sources integration possible. Ontology-to-data source mapping will add semantic description layer to each of ISs involved in the process.

Developing a semantic layer related to a single data source gives the possibility of a rapid inclusion of the observed sources in the community of semantically integrated data sources. The aim of this paper is to present a solution that performs automation of a single data source into a community of semantically integrated data sources inclusion process. The rest of the work will be organized in the following way. The second chapter describes the current research in the area of information system interoperability and development based on the ontology usage. The third chapter presents the parts of architecture that allow automatic data access layer generation. It also describes the possibilities that this layer should provide. The fourth chapter contains the advantages and disadvantages proposed by the solution as well as planned future development.

RELATED WORK

An opportunity to take full advantage of the information integration is based on utilization of data semantics and the context that the data is used in. Most frequently used way for presenting connotation, (e.g. obtained information semantics) relies on the ontology usage. The term "ontology" (Guarino, 1998) is used in many different contexts, but the instance of certain ontology type is mainly a consequence of conceptual analysis of certain parts of reality that is significant to information system developers. Characteristics of this part of reality are represented by data instances contained in IS data sources. Both ontology and ontology instance can also be a product of the modeling process of certain interest domain (Fonseca and Egenhofer, 1999). Analysis and modeling processes are often followed by ontology creation and instantiation.

One of the ways to prepare IS for a quick and easy inclusion into the integrated IS community is to observe ontology as a baseline model in the IS construction process. Observed in this role, the ontology would be moved to the center of the IS life cycle, thus making this IS an Ontology-Driven IS (Fonseca and Egenhofer, 1999). If IS developers' goal is to achieve interoperability and include the new IS into the existing IS community, then ontology should be used as a baseline model for the system development. This baseline model represents the specialization of a common model used in the community that the observed IS needs to be added to. A priori possession of a baseline model should automate and accelerate the development of interoperable IS.

Information integration demand three types of heterogeneity to be solved: semantic, schematic and syntax heterogeneity (Fonseca et al, 2000). Semantic heterogeneity is especially emphasized in cases when many user communities exist in same domain. Semantic heterogeneity problems can be overcome by introducing standards (Bishr, 1997) to be applied to the community. A more efficient, and in the past broadly used approach is based on the use of semantic translators that are introduced with the aim of providing interoperability of the user communities. Good examples of the semantic translator based approaches are GeoNis (Stoimenov and Đorđević-Kajan, 2004) and Buster (Visser, 2004).

One of the side effects of information systems implementation and use is the creation of user communities. In the beginning user communities had a predefined set of functionality and data. Today, user communities have evolved into groups that, while using the IS, affect its characteristics, change collection of data that IS uses and meaning of information available in the IS. It is possible to classify communities based on the domain they belong to. Each community has information related to the observed domain and its activity creates a subset of information that relies on the conceptualization used in the process of a single IS development.

Although being very complicated in structure, geographic and spatial data domain can have a large number of users. Geoinformation Community (GIC) is the term widely used for the community that these users form. Immense GI system popularity is the consequence of the possibility to simply combine spatial data with data from other domains. Data from other domains are often joined to geographic objects as thematic attributes. Thematic attributes of geographic objects can be presented as separate concepts in non-geographic domains. This way information representation provides an opportunity to combine and integrate information from different domains.

IS domain conceptualizations are usually represented by application ontologies that can be mutually connected if there is a mapping between them. In addition to application ontology mapping, application ontology linking can be made using common domain ontology. The combination of these two approaches is referred to as a hybrid approach (Stoimenov and Đorđević-Kajan, 2003). Application and domain ontology usage allows combining information from different sources. The starting point of this approach is to represent ontology in the form of a dynamic object-oriented structure (Fonseca and Egenhofer, 1999). Presented in this manner, ontology will inherit the characteristics of the domain object model that user community is connected to, e.g. the characteristics of information system user community uses. Therefore, it is possible to apply the agile methodology data source mapping to this ontology.

Infrastructure for the semantic information integration can be realized using different approaches. For example, architectures proposed by (Stoimenov and Đorđević-Kajan, 2004) and (Visser, 2004) are based on hybrid ontology approach. This approach is based on the use of ontologies with different level of generality (Guarino, 1998). Figure 1 represents a typical overall architecture that solves problem of semantic heterogeneity. This architecture clearly separates three components that interact in the process of information exchange: Data users, Semantic layer and Geoinformation Community.

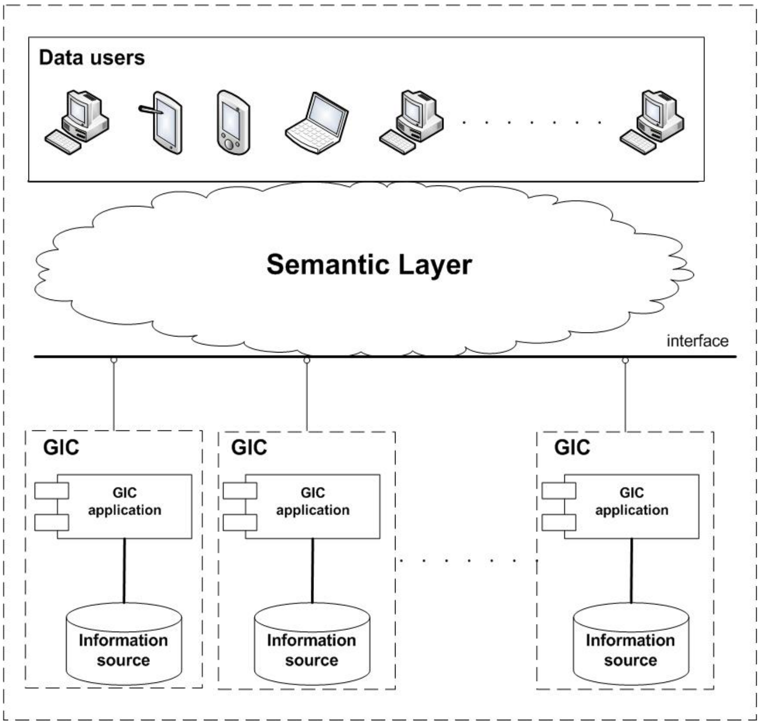

Figure 1: GIC and interoperable environment.

Regardless of the approach used for ontology development, common characteristic is the need to provide interfaces to various GIC that may appear in the domain. Number of GICs that will participate in the process is not known in advance and may change in time, e.g. some GICs leave the process of information exchange and new ones appear and start the process. Due to the dynamic nature of communities that participate in the process of information exchange, it is necessary to provide mechanisms that will provide a simple and effective way for new interested users and user communities to be included in the process. This is particularly true when adding new data sources into semantic information integration infrastructure. Mostly, infrastructures for semantic information integration expect information sources to offer their services using a uniform interface. As a result, when a new data source is added into the infrastructure, there is a need for implementation of components that will provide the interface and will enable the flow of information to and from semantic interoperability infrastructure.

DATA ACCESS LAYER AUTOMATIC GENERATION

If each of the domains has a semantic description (domain ontologies), it is possible to achieve semantic information integration of the domain's ISs. If semantic information integration is to be observed in the domain of GI systems and GICs with commitment on the unique ontology, joining the new GIC to the existing community could be fully automated. Aim of this research is development of a tool that would automate process of joining new GICs to an existing interoperable environment. This tool would use ontology, the layer that describes ontology-to-relational database mapping and the relational database scheme to generate data manipulation functions. Prerequisite for this automation is the existence of the above-mentioned intermediate layer between ontologies and relational database content. Such tool would enable quick and easy inclusion of GIC into the existing GIC community and automate the process of semantic integration of new GIC information with information from the community.

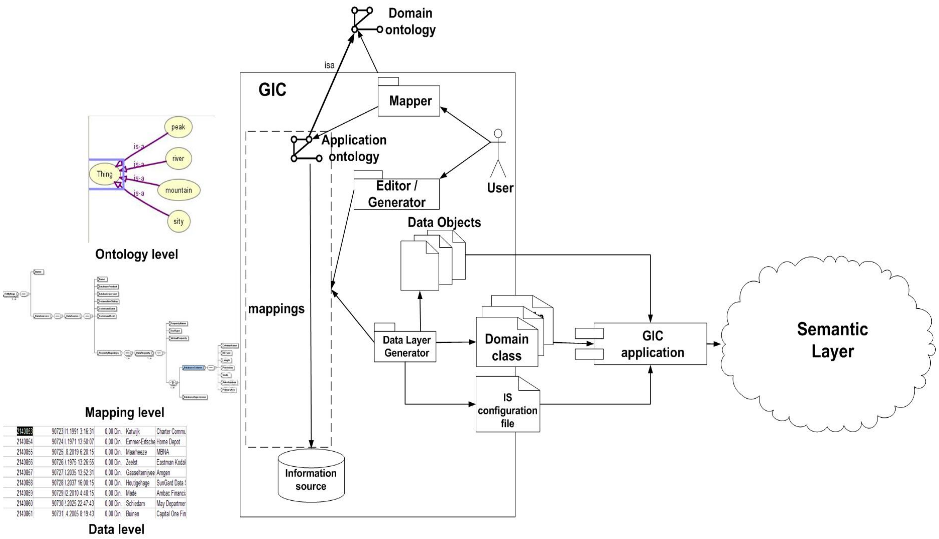

Figure 2: Data Access Layer Generator in GIC environment.

In addition, there is a need to completely integrate information sources related to the individual GIC-s. Figure 2 shows position of Data Access Layer generator in a GIC environment. Information integration includes a development of a metadata layer for each of GIC-s based on application ontology-to-concrete information source mapping. This metadata includes ontology-to-RDB mappings and domain objects-to-information sources mappings. On the basis of this metadata, it is possible to create a component that automates the process of GIC data layer development. Data layer is a key component necessary for the creation of GIC applications. GIC application has the task to link data sources that GIC possesses with the semantic integration infrastructure. Semantic infrastructure defines a standard linking interface which enables communication in both directions. If the observed domain is GI domain, this standard interface is usually based on WFS specification (OGC, 2002). Implementation of these interfaces is permanent in most cases while the only critical part is the part that works with the data because it changes from one to another GIC environment since information sources change. Therefore, a component that enables the automatic data layer creation would significantly speed up the GIC application development process. The generated data layer would be added to standard GIC application interface implementation. This process would further enable GIC inclusion into semantic interoperability infrastructure. In the following sections the key parts of data layer automatic generation architecture will be displayed.

OWL2RDB Mapping Schema

The creation of semantic description of the data, generated and stored by legacy systems, is a very well known problem in computer science community. The majority of legacy systems are based on the usage of relational database technology and for that reason the development of semantically based relational database – to – ontology mediation layer is often suggested (Konstantinou et al, 2004). This layer is the essential prerequisite for the usage of data from the relational databases (RDB). It usually consists of semantically based relational database-to-ontology mapping files, created using various languages.

Ontology-to-RDB mapping languages are specially developed for the purpose of describing relational database content. The problems these languages deal with arise from the low similarity between the ontology and RDB models (Barrasa et al., 2008). The overall interest in mapping ontologies to relational databases caused a number of mapping approaches to appear. World Wide Web Consortium (W3C) formed an incubator group (W3C RDB2RDF Incubator Group, 2008) to examine and classify the existing approaches to mapping relational data into RDF/OWL and decide whether standardization is possible and/or necessary. The language used for ontology creation is in most cases RDF(S) (W3C, 2004a) or OWL (W3C, 2004b).

OWL2RDB mapping schema is the corner stone of semantically based relational database – to ontology mediation layer used by Data Access Layer Generator. Due to OWL2RDB mapping schema importance a lot of attention was devoted to the process of mapping ontology concepts and data properties. For development of our OWL2RDB mapping schema we used approach suggested in R2O mapping language (Barrasa et. al, 2004).

For the purpose of clarity and understanding, RDB tables and columns will be referred to as "mapped" elements while ontology concepts and the properties will be referred to as "mapping" elements. Mapping file consists of a series of EntityMap elements as shown in Figure 3. Each EntityMap element contains a collection of DataSources elements that define data sources which can be used in order to instantiate mapped concepts. Each DataSource element is one of the data sources (tables, views, stored procedures, SQL command...) that the observed ontology concept can be mapped to. DataSource element description contains information necessary to access data in the concrete RDB: database product, version, connection string as well as a description of the mechanism that is used to access the information (whether the table is accessed directly or executing SQL command that fetches data).

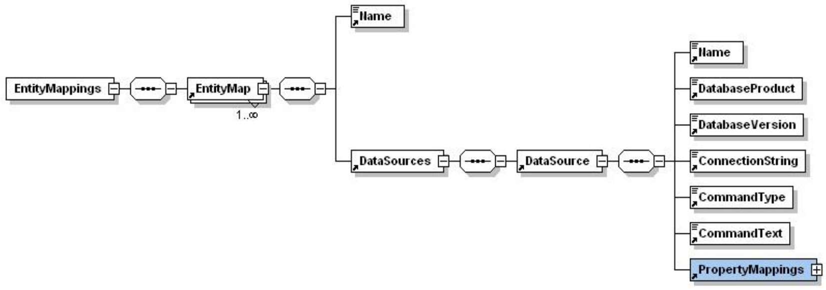

Figure 3: EntityMappings in OWL2RDB mapping schema.

PropertyMapping element is a collection of DataProperty elements that define the mapping between the ontology data properties and data in RDB (Figure 4). Each DataProperty describes ontology data property, whose domain includes the observed ontology concept. DataProperty element is associated with a particular relation column (relations that physically exist or are the result of some SQL commands) to which the observed ontology concept is mapped (DatabaseColumn element). For each ontology data property, mapping file has a property name (PropertyName element) and OWL type (OwlType element). For the appropriate table or relation column that the property data property is mapped to, mapping file has a column name (ColumnName element) and column type (DbType element). Optionally, DatabaseColumn element can contain additional elements (e.g. AutoNumber, PrimaryKey …) in order to represent some specifics of selected RDB technology.

DataProperty element can also contain VirtualProperty element. This element describes RDB relation columns that can not be mapped to the ontology properties. These RDB relation columns do not have matching ontology data properties. Necessity of their virtual mapping arises from implementation details and information described by these columns is sometimes referred to as "shadow information" (Ambler, 2002a).

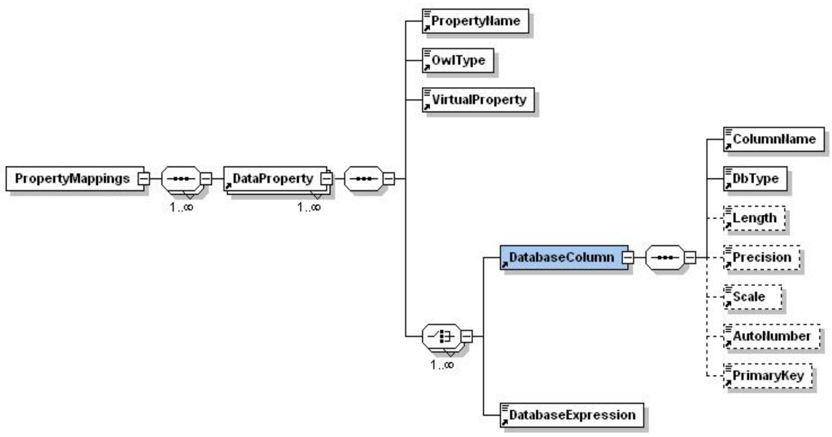

Figure 4: PropertyMappings in OWL2RDB mapping schema.

Data Access Layer Generator

Information integration used by individual GIC facilities (Figure 1) relies on the use of application ontology instance that are related to individual information sources. The process of application ontology instance development can be fully automated by using Data Access Layer Generator.

Data Access Layer Generator implies the existence of application ontology and application ontology-to-data sources mapping file (in this case, application ontology-to-RDB mapping file). Application ontology can be observed as object - oriented structure (Fonseca and Egenhofer, 1999) and the problem of creating instances of application ontology can be seen as a problem of creating domain objects based on the data in the system. Most of GIC facilities used relational database technology for storing information. Problem known as “object-relational impedance mismatch” has appeared as a result of combining object-oriented and relational data models (Ambler, 2002a).

Techniques for overcoming the problem of object-oriented impedance mismatch are discussed in detail in the agile methodology for software and databases development (Ambler, 2002c). This methodology suggests usage of object-relational mapping technology as a solution for the problem of object-oriented impedance mismatch. This technology allows direct relation attribute-to- class attributes mapping and solves problems of modeling object behavior in relation to data they use.

O/R mapping, as suggested in agile methodology, also deals with any data objects that need to persist to maintain themselves. Additional information used for object persistence represents "shadow information" (Ambler, 2002b) and is very important for the system implementation.

Object – to - relation mapping is a technology that overcomes some of the mismatches that exist between the object-oriented and relational paradigm. As a result, the popularity of the proposed technologies increases. Hence, a large number of ready-made tools (O/R mapping tools) are present on the market. These tools allow mapping between objects and RDB. Examples of these tools are Zend (Zend 2006), Hibernate/NHibernate (Hibernate 2008), etc.

By representing ontology in the form of an object-oriented structure, ontology concepts can be observed as domain objects that can be mapped to RDB data. Therefore, in architecture shown on Figure 2, O/R mapping tool has an important role in the process of automatic Data Access Layer generation.

For the purposes of automation of the application ontology instances and GIC applications generation process, a special tool is developed – Data Access Layer Generator. The functioning of this tool is based on mappings described in the previous two chapters: ontology – to - RDB mapping layer and domain objects – to – RDB data mapping. In order to create ontology – to – RDB mapping, the OWL2RDB mapping scheme was used. For the purposes of mapping domain objects to RDB data, we used the Hibernate/NHbernate (Hibernate, 2008) implementation of O/R mapping technology. The process of creating application ontology instances and GIC applications is shown in Figure 5.

Data Access Layer Generator parses application ontology distinguishing concepts, data and object properties. At the end of this phase, a single class (source code) is generated for each of the selected concepts. Based on their domain attributes, for every data property class member attribute (with same name and proper data type) is generated. For example, let us consider application ontology that contains height data property and peak and mountain concepts. Height data property has domain attribute which contain peak and mountain ontology concepts. Classes (source code) generated for peak and mountain ontology concepts will then have height class member attribute whose type will correspond to height ontology data property type.

Ontology object properties are considered if they reflect the aggregation (whole-part association) of ontology concepts. In Ontology Web Language (OWL), this type of association is presented using the keyword "has". For example, let us consider application ontology that contains mountain and peak concepts and has some peak object property. If has some peak object property has mountain as its domain attribute and peak as its range attribute, then Data Access Layer Generator will add a list of object references into the class generated for the mountain ontology concept and these references will point to object instances of class generated for peak ontology concept.

After the creation of object-oriented application ontology representation, e.g. after the creation of classes used for a concept, data and object properties modeling, elements (source code, mappings and configuration files) required for Hibernate/NHibernate O/R mapping tool usage are created:

- *Domain object (classes)* – instances of these classes represent domain object used by O/R mapping tool (DomainObject component in O/R mapping pattern)
- *Hibernate/NHibernate hbm files* – these files represent MapMetadata O/R mapping tool files (MapMetadata utility component in O/R mapping pattern)
- *Application configuration* – configuration parameters used for GIC application generation (list of all concepts that that domain objects are generated for, domain object hierarchy, Hibernate/Nhibernate component enablement parameters - configuration for PhysicalDatabaseDriver component in O/R mapping pattern)
- *Physical database driver configuration* – necessary pieces of information in order to connect to data source

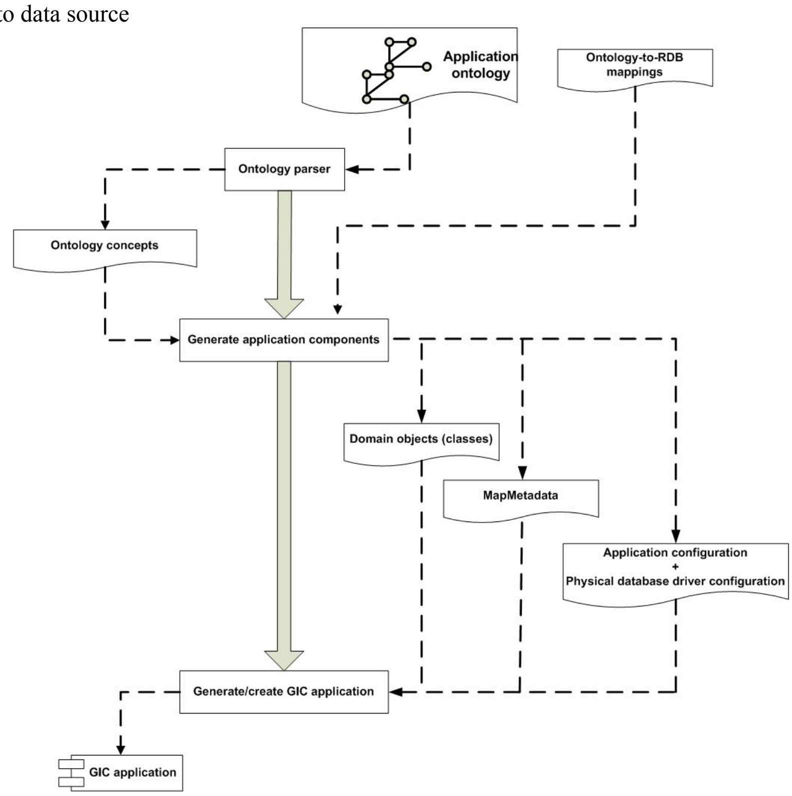

Figure 5: Process of GIC application development.

Each of the GIC application has the possibility to include the components created by Data Access Layer Generator tool. Starting from the application ontology, GIC applications are provided with an opportunity to generate an object-oriented representation of application ontology instance using the generated components. Each application has the possibility of instantiating application ontology as a collection of objects that represent an instance of application ontology concepts. Inter-connection of instantiated objects matches the connections between concepts as presented in application ontology. By placing the generated components in the existing GIC application, their functionalities are expanded in terms of information sharing and integration. GIC applications developed in this manner are the basis for achieving interoperability of a GI community that uses common domain ontology.

CONCLUSION

Interoperability of information systems basically represents the possibility of individual systems to exchange information and use the exchanged information. If any data acquisition is to be possible, each of the information sources has to provide an interface known to other information exchange participants. Also, it is necessary to understand the format of the acquired data in order to interpret them correctly. These requirements implement the minimum that has to be fulfilled before considering the acquired information semantics.

The consideration of integrated information meaning introduces additional quality to the way semantically integrated information systems function. The starting point in achieving this level of quality of IS functioning is the creation of a semantic description of the information sources related to the observed system. For this purpose, ontologies are the most frequently used technique. Their overall usage influenced the change of the role that ontologies play in modern information systems. From information semantics description, through means to achieve semantic interoperability of an isolated IS, ontologies have taken a central place in the process of IS construction, use and integration.

Represented as an object-oriented structure, ontology is an equally effective means for achieving interoperability of the existing (legacy) system. If the goal is to quickly include the existing IS into semantic interoperability infrastructure, representing IS semantics in the form of object-oriented structure maps the problem of semantic integration to the problem of mapping objects to information sources. If the legacy IS uses RDB as its information source, which is the case in most situations, the problem boils down to mapping object-oriented IS semantics representation to relational database data. An approach presented in this paper offers a solution to this problem and the solution to the problem of including the existing IS into semantic interoperability infrastructure in a fast and efficient way.

The problem of semantic information integration and inclusion of the existing IS into semantic interoperability infrastructure is present and discussed in the field of GIS-a. Inclusion of the existing GIS applications into semantic interoperability infrastructure provides users gathered around observed applications (GIC) an opportunity to use information from various sources.

BIBLIOGRAPHY

Ambler S. W. (2002a) The Object-Relational Impedance Mismatch, http://www.agiledata.org/essays/impedanceMismatch.html, Last date accessed 12.2008.

Ambler S. W. (2002b) Mapping Objects to Relational Databases: O/R Mapping In Detail, http://www.agiledata.org/essays/mappingObjects.html, Last date accessed 12.2008.

Ambler S. W. (2002c) Techniques for Successful Evolutionary/Agile Database Development, http://www.agiledata.org/, Last date accessed 12.2008.

Ambler S. W. (2003) Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development, http://www.agilemodeling.com/essays/amdd.htm, Last date accessed 12.2008.

Barrasa J., Corcho O., Gomez-Perez A. (2004) R2O, an Extensible and Semantically Based Database-to-ontology Mapping Language, In Proceedings of the 2nd Workshop on Semantic Web and Databases, Toronto, Canada

Bishr, Y. (1997) Semantic Aspect of Interoperable GIS, Wageningen Agricultural University ITC, pp. 154

Fonseca F.T., Egenhofer M.J. (1999) Ontology-Driven Geographic Information System, 7th ACM Symposium on Advances in Geographic Information System, Kansas City

Fonseca F.T., Egenhofer M.J., Davis Jr. C. A., Borges K. A. V. (2000) Ontologies and Knowledge Sharing in Urban GIS, Computer, Environment and Urban Systems, vol. 24, pp. 251-271

Guarino N. (1998) Formal Ontology and Information Systems, In N. Guarino (ed.) Formal Ontology and Information Systems. Proceedings of FOIS'98, Trento, Italy, IOS Press, Amsterdam, pp.3-15.

Hibernate (2008) NHibernate 2.0.1GA, http://www.hibernate.org/, Last date accessed 12.2008.

Konstantinou N., Spanos D.-E., Chalas M., Solidakis E., Mitrou N. (2006) VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Content, Web Information System Modeling, pp. 1050-1061

Open Geospatial Consortium (2002) Web Feature Server Interface Implementation Specification, Version 1.0.0, OpenGIS Project

OWLDOTNETAPI (2006) http://users.skynet.be/bpellens/OwlDotNetApi/index.html, Last date accessed 12.2008.

Protégé (2004), http://protege.stanford.edu/, Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine Last date accessed 12.2008.

Stoimenov L., Đorđević-Kajan S. (2004) An architecture for interoperable GIS use in a local community environment, Computers & Geosicence, Elsevier, Volume 31, Issue 2, March 2005, Pages 211-220

Stoimenov L., Đorđević-Kajan S. (2003) Realization of GIS Semantic Interoperability in Local Community Environment", Proceedings printed as book, Eds. M.Gould, R.Laurini, S.Coulderon, ISBN 2-88074-541-1, 2003, Presses Polytecniques et Universitares Romandes, 6th AGILE conference on Geographic Information Science, "The Science behind the Infrastructure", AGILE 2003, Lion, France, April 20-23.2003. pp.73-80

Visser U. (2004) The Buster Approach for Terminological, Spatial and Temporal Representation and Reasoning, ISBN 978-3-540-22993-3, Springer Berlin / Heidelberg

Zend (2006) http://framework.zend.com/, Last date accessed 12.2008.

W3C (2008) RDB2RDF Incubator Group, http://www.w3.org/2005/Incubator/rdb2rdf/, Last date accessed 12.2008.

W3C (2004a) Resource Description Framework (RDF), http://www.w3.org/RDF/, Last date accessed 12.2008.

W3C (2004b) OWL Ontology Web Language, http://www.w3.org/2004/OWL/, Last date accessed 12.2008.