AN ACTIVE, OBJECT-ORIENTED MEDIATOR-BASED FRAMEWORK FOR THE DEVELOPMENT OF GIS APPLICATIONS

Leonid STOIMENOV, Dragan STOJANOVIĆ,
Department of Computer Science, Faculty of Electronic Engineering,
University of Niš, Beogradska 14, 18000 Niš, tel. +381(0)18 529-331, E-mail: leni@elfak.ni.ac.yu

SUMMARY

Most current geographic information systems assume and present the static world. But, new generations of GIS applications have much more demands in comparison to possibilities, which could provide traditional database management systems. Information that exists in the spatial database may be updated over time. Also, the very important demand is time capability to answer to state changing in database.

The goals of our research activities are defining an active architecture, identifying the need for mediators in GIS, adding the active rules to the GinisNT spatial database and realize the active mediator level which perform active behaviour of the GinisNT. Paper describes how complete active database semantics can be supported on an existing framework by adding a mediator between the GinisNT spatial database and the client applications. ECA rules are fully supported through the ActiveMediator component without changing applications or the spatial database schema. The Active Mediator prototype provides integration of active and object-oriented features in one system. To provide support for temporal GIS application design and implementation, a suite of spatio-temporal object modelling and management technologies and tools has been developed constituting spatio-temporal object modelling and management system. The openness and scalability of developed frameworks enable their further improvement and refinement to satisfy the needs of specific, temporal GIS applications. Active, temporal GIS would improve understanding of the dynamic geographic processes caused by man, the nature or both, provide methods for detecting and analyzing trends and cycles in geographic phenomena, an would enable prediction of the future geographic states.

Keywords: *GIS, mediator, active databases, temporal databases.*

1. INDRODUCTION

Geographic information systems are computerized systems for managing data about spatially referenced objects. GIS differ from other types of information systems in that they manage huge quantities of data, require complex concepts to describe the geometry of objects and specify complex topological relationships between them [11]. In addition, GIS data are typically used by various groups of users with different views and needs. Most current geographic information systems assume and present the static world. But, new generations of GIS applications have much more demands in comparison to possibilities, which could provide traditional database management systems. Information that exists in the spatial database may be updated over time. The very important demand is time capability to answer to state changing in database [1].

To support applications such as GIS, database management systems (DBMS) need to have some support for time. This limitation of current GIS capabilities has recently become the focus of growing research interests within GIS community [2]. Solution for these demands is the active [1,6] and temporal database systems [14,3]. In an active DBMS it is desirable have active rules that can access the time when event occurred. A temporal DBMS usually has extensions to the basic relational operations to support queries that use the time dimension.

The growing demand for better understanding and analysis of everlasting geographic changes has emphasized the need for GISs with temporal and active capabilities developed around spatio-temporal database system [9]. Applications in domains so diverse as earth sciences, economical and socio-economical studies, urban planning, traffic control, land management, environment protection, medical imaging, or fleet tracking, can benefit from integration of spatial and temporal information. Temporal GIS could be used to more closely depict the knowledge of the real world, to better explain dynamic geographic processes and phenomena, reveal patterns in their occurrence and explore cause and effect relationships between them. This caused the integration of research work and technology of spatial and temporal databases and resulted in design and development of spatio-temporal data models, database management systems and information systems. Active, temporal GIS would improve understanding of the dynamic geographic processes caused by man, the nature or both, provide methods for detecting and analyzing trends and cycles in geographic phenomena, an would enable prediction of the future geographic states.

The research group at the Computer Graphics and GIS Lab at the University of Nis, Yugoslavia, has been developing GIS software for ten years now. One of the research directions pursued has been the development of GIS architecture suitable for the implementation of end-user GIS applications under very limited resources. This paper presents our approach to the building active layer under an existing OO GIS environment called GinisNT [11]. Also, aim of this research is to bridge the gap between research and implementation in temporal GIS domain by development of temporal GIS application framework based on spatio-temporal object modeling and management system.

The paper is structured as follows. Section 2 of this paper describes basic capabilities of active database systems, related work on mediators, temporal data models and temporal databases. Section 3 presents the architecture, design, and some implementation details of GinisNT and discusses its components and active and temporal extensions. This section contains details of the Active Mediator prototype, which provides the active functionality for the GinisNT environment. The approach is called Active Mediator, since it introduces an intermediate level of mediator software between data sources and their use in applications and by users, and since it supports active database facilities. Section 4 (Conclusion) describes advantages of this approach, and gives future research directions.

2. RELATED WORK

During the last decade intensive research in spatial data models, database management systems (DBMS) and database applications has been performed within a database community, mostly independently. Spatial database research has focused on modeling, storing, querying and integrating geometric and topological information in databases [8]. Early geographic information systems (GIS) stored spatial data using dedicated data structures within file systems with connection to thematic information stored in a standard relational database. The advent of database technology caused that further research was focused on integration of spatial and thematic data either within pure relational models and database systems, extended relational database systems supporting additional data types and index structures [23], or more recently within object-oriented data models and database systems [7].

Many GIS applications use the concept of time. To support applications such as GIS, DBMS need to have some support for time. Traditional DBMS are passive because application program initiated operations or user explicitly invokes them. Applications send request for operations to be performed by the DBMS and wait for answers. However, active database management systems (ADBMS) are event driven systems where changes in database can be monitored by active rules. An ADBMS operation can be invoked, not only by events that have been generated by users or application programs, but also by external events such as changes of input values or time event. Temporal DBMS usually has extensions to the basic relational operations to support queries that use the time dimension.

2.1. Active databases

ADBMS have been developed to support applications with detecting changes in database [1]. It is able to detect specific situations in database and to perform corresponding actions specified by the user. This includes support for specifying active rules that monitor changes to data, and rules that perform some control tasks for the applications.

Active database systems are primarily database management systems with the main task of storing large amounts of data and providing efficient access through a query language. The ADBMS is a DBMS, and all the concepts required for the passive DBMS are required for the ADBMS. If the user ignores active functionality, the ADBMS can be used in the same way as a passive DBMS [1]. Active databases provide enhancement of passive database features, in order to:

1. React on changes in the status of objects managed within the database, or
2. React on time events, or
3. React on other application specific events.

Required functionality for an ADBMS includes support for creating, modifying, activating and deactivating ECA (Event Condition Action) rules [6]. The ADBMS must support event monitoring and storing events in an event history [4].

Active rules (ECA rules) are primarily used for monitoring changes in database. The rules can directly access data stored in the database. According to the traditional coding techniques, where functionality of the system is in the modules and functions, the active rules provide more dynamic way of handling new situations and introduce the active behavior into the system. While a number of research prototypes of active database systems have been built, ECA rule capability in Relational DBMSs is still very limited. Currently, commercial object relational databases do not support composite events. However, several of the more advanced ECA rule features can be added to an existing DBMS. For example, Li and Chakravarthy [10] recently added an ECA rule based mediator between the SQL Server and the clients.

Two different approaches have been distinguished in incorporating of active behavior in databases [6]. The first one (used in our GinisNT framework) is layer architecture, which makes a new layer on-top existing database. The new layer implements the active behavior and database is treated as a black box for active layer. The second approach is integrated architecture, which means internal change of DBMS kernel and using new systems/prototypes. However, it makes leaving existing systems and resources.

The better approach is first one, i.e. to expand existing database system with active behavior. The advantages of the layer architecture are reduced time of implementation and the possibilities of using the existing database resources. The complete active database semantics can be supported on an existing database management system by adding a intermediate layer (called mediator, described in the next section of this paper) between the DBMS and the client applications. AMOS (Active Mediators Object System) is an example of layered architecture what model, locate, search, combine, update, and monitor data in information systems [5].

2.2. Mediation

Information mediators are originally developed for integrating information in databases. Wiederhold defines a mediator in [25] and he is the first that point out the need of mediation for contemporary database systems. Wiederhold therefore sees mediator as an intermediate abstraction layer between databases and applications that use them. Mediation is primary an architectural concept therefore the precise implementation of a module is less important. The mediator architecture is therefore introduced as a three layers system: (1) applications, (2) mediators, and (3) DBMS. The three-layer architecture of mediator-based systems includes a heterogeneous databases with wrappers (DBMS layer), a mediation layer, which supports exchange of queries and results between wrapped legacy data sources and applications, and an application/user interface layer [24]. A wrapper is a program that is specific to every data source [13]. Wrapper extracts a set of tuples from source file and performs translation in the data source format. Mediator architecture provides transparent view of data sources and independence of data sources and user applications [18].

Mediators are not just simple interface between applications and databases but they have to include some knowledge in themselves that cannot exist in data they work on. They have to aggregate underlying data depending on criteria dictated by the application layer. The process of understanding which domain contains the best information for answering a query is delegated to the automatic knowledge based engines built in the mediators themselves. Mediators are a good way for integrating completely different data source types [17]. The most important fact is that data integration system lets users focus on specifying what they want rather than thinking about how to obtain the answers. As a result, it frees them of combining data from multiple sources, interacting with each source and finding the relevant sources.

2.3. Temporal database research

Temporal database research has concentrated on modeling, querying and recording temporal evolution of real world information through different types of time (valid, transaction) and thus, on extending the current information, modeled and stored in databases, toward past and future domains [21]. The integration of time in databases systems has been an active field of research since the mid 80's. Because of dominance and wide acceptance of the relational data model, most research was dedicated to development of temporal extension of relational data model using either tuple or attribute timestamping [14]. With the advent of object-oriented computing paradigm, the emphasis of research in temporal database field was also put on integration of temporal information in object-relational and object-oriented data models and database systems [15]. Such research was followed by development of appropriate temporal query languages [22]. However, very little of the mentioned work had an impact on software providers, and current commercial DBMSs are only beginning to offer some rudimentary time-related functions, such as Oracle 8i Time Series.

The spatio-temporal data models reviewed in literature are specified on conceptual and logical levels. Mostly, the term data model doesn't only refer to the data model of the underlying database system (e.g., relational, object-relational or object-oriented), but to the comprehensive principles how spatial, temporal and thematic data are modeled and represented in the information system, both within GIS application and underlying database system. Comprehensive survey and classification of such models are presented in [12], with critical evaluation of different approaches through use of a case study and the construction of a comparison framework. The general conclusion is that majority of reviewed data models are more focused towards modeling, and less to data management and data model implementation. Either these data models seem to be unfeasible to implement or have to be changed significantly to fit into the architectural design of commonly available database systems and application development frameworks.

3. FOUNDATION OF GinisNT

One of the inspiring factors for this research was the inadequacy of classical GIS architectures and commercially available GIS environments [11]. They are invariably developed with some sort of spatial applications in mind, so only certain spatial data types and operations are predefined and the tools for end-user application development are not flexible enough. Most of these environments are closed; there is no opportunity for the application developer to extend them and add some desired functionality. Therefore, the development of a spatial application can be very difficult if it is substantially different from those supported by the environment used.

The other factor was the resources that were available. In developing countries such as Yugoslavia, not only funds are highly restricted, but also there are additional problems, like inadequate infrastructure, instability and lack of educated people. In such settings, GIS software must run on cheap hardware platforms, which may even be shared with other software. It is often the case that a single computer will be used for a GIS application and also for some other applications, like resource management. Furthermore, data should be easily interchangeable between GISs and other applications.

New spatial data sources (GPS measurements, satellite remote sensed data, available spatial databases) should be used for data acquisition. The level of data accuracy should be chosen as to satisfy the requirements of applications, but not to impose too high a demand on existing resources, since increasing data accuracy will result in exponential increase of required resources.

3.4. GinisNT object model

We have developed our own OO model embedded into GinisNT, based on the concepts found commonly in the literature. GinisNT object data model supports objects, classes, attributes, object identity (OID), aggregation, generalization/specialization and associations. An object represents a real-world entity, which is uniquely identifiable by means of OID. A class describes all the objects with the same structure and behavior. A class is a template for a set of objects, and the objects are instantiations of a class. Classes can be ordered by generalization/specialization relationships, thus forming class hierarchies and enabling subclasses to inherit attributes and methods from superclasses.

Each object possesses a set of methods and attributes. The value of an attribute can be a data item, an object, or even a set of objects. This characteristic allows the definition of composite objects as the aggregation of simple objects. Each component of a composite object is thus represented as an embedded object. Aggregation models the whole-part relationships between objects and can be multi-layered and recursive. Associations model relationships that exist between various objects. Associations are explicitly specified on the OO schema, while the relational schemas represent them by means of foreign keys.

Users define applications by developing their OO models. In order to do that, users select classes from a predefined class library and adapt them to the application's needs, or develop completely new classes with no relations to other classes in the library.

The GinisNT class library consists of four groups of spatial and non-spatial classes. The first group contains the classes used in the metadata repository, which correspond to the concepts of the underlying OO model. Another part of the class library is a set of project definition classes, which hold general information about the project, such as the date of creation, the name(s) of the designer, purpose and similar. Metadata and project classes are not extensible. The class library also contains an extensible hierarchy of attributes and an extensible feature hierarchy. The feature hierarchy contains both spatial and non-spatial classes. GinisNT feature hierarchy provides definitions of features representing the primitive, raster and vector, spatial objects such as point, line, area, point-array, point-matrix, etc., and node, chain, polygon and DEM topological ones. Its root is the GNTFeature class, which provides object identification and the basic support for non-spatial classes. Spatial classes make the bulk of this hierarchy, and are organised in three layers: geometrical, describing the shape of spatial objects, topological, describing relations of a spatial objects to other objects, and user-defined. These classes contain a description of geometry and attributes related to spatial abstraction, and a set of appropriate geometrical and topological operations for processing of this data. Extensibility of feature hierarchy enabling the user to specify application specific classes with inherited geometry and related operations, and defined special properties and behavior of the real world entity [19].

3.5. Temporal extension of GinisNT object model

In order to suitable perform task of developing GIS applications that need temporal dimensions of spatial data to be maintained and managed, GinisNT must be extended with temporal capabilities. The extensible class library is the starting point in developing temporal GinisNT extension. GinisNT temporal extension [20] is based on temporal hierarchy with class Temporal feature as a top-level class, and multiple inheritance as one of the core concepts of object-oriented paradigm. Temporal classes for time instant, time period, time duration, and complex time object are specialized from the base. The class definitions include appropriate temporal data such as year, month, day, hour, minute, seconds, etc, temporal reference system, as well as temporal topology operators and general purpose temporal functions. The class of particular spatiotemporal object is specified via multiple inheritance from appropriate feature class describing its spatial properties and the temporal class describing its temporal characteristic. Thus, according to the mechanism of object-oriented inheritance, the spatial and temporal dimensions are integrated within single class abstraction in addition to concentrate thematic attributes and operations specified for that class of spatiotemporal objects.

Temporal extension of GinisNT is based on the generic, object-oriented spatio-temporal-thematic data model named STOM (Spatio-Temporal Object Model) [20].

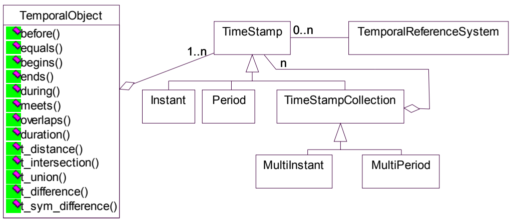

Fig. 1 ***TemporalObject*** class and ***TimeStamp*** class hierarchy.

In our object-oriented spatio-temporal data model we have proposed a discrete model of time, and object-timestamping by valid time. A temporal object is a real or abstract feature from the real world, which has or may potentially have an associated temporal property indicating its position (existence) in time, time of change of its properties and duration of existence. Temporal properties of features modeled in STOM are represented using TemporalObject class. Time dimension of temporal features is specified through TimeStamp class hierarchy. It defines primitive temporal classes which can be used for timestamping objects: instant (Instant class), period (Period class), and homogenous (MultiInstant and MultiPeriod classes) and heterogeneous (TimeStampCollection class) collections of time stamps (figure 1). TimeStamp class is associated to TemporalReferenceSytem class, which defines time type (UTC, GPS or arbitrary time), and an offset from UTC time in hours and minutes.

Active behavior is represented through classes CActiveObject and CActiveAttribute. Classes of objects that are considered as active must be defined through inheritance of the CActiveObject class. Passive classess may have some attributes with active behaviour. In that case, active attributes is declared as CActiveAtribute type.

4. ARCHITECTURE OF GinisNT

The GinisNT is an OO platform for the development of scalable, end-user GIS applications. It is based on a relational DBMS at implementational level, the usage of which is made transparent to the user completely [18]. This transparency is achieved by using a mapping algorithm implemented within the Mediator [18,19], one of the components of GinisNT. GinisNT provides the basic GIS functionality found in commercial GIS software, with the additional advantages of flexibility and the ease of development of end-user applications. The architecture of GinisNT with Active Mediator layer is shown on Figure 2.

ObjectWizard is an OO CASE tool in which end-users can develop OO applications. ObjectWizard supports modelling phase, performs automatic mapping of the OO model of application into the relational schema, creates all the necessary structures in relational database and also stores definition of the application in the metadata repository. User-defined schemas are stored in the Metadata Repository (MDR). The existence of MDR frees a user from the burden of memorising database schemas and corresponding access mechanisms. ObjectWizard stores definitions of classes and corresponding information about tables in MDR when the classes are mapped into the relational data model. MDR also supports schema evolution. Users can easily change schemas even without changing underlying code manually.

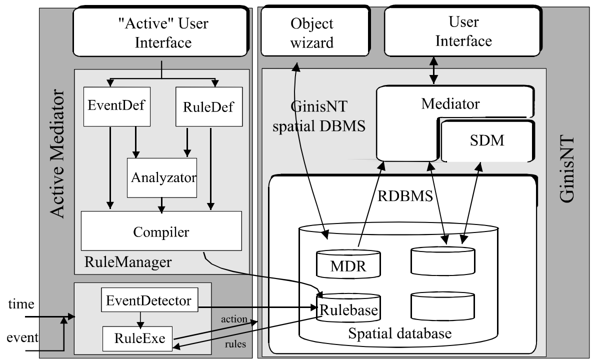

Fig. 2 Architecture of GinisNT with Active Mediator layer.

The user interface provides means for communication between end-users and their applications, including the functions for displaying data (both attribute and spatial data). The user interface is configured automatically using information from MDR.

Mediator is the most important component of this architecture. It interfaces end-user, metadata repository and the database. Mediator manages attribute data (alphanumeric and other descriptive data), while spatial data is handled exclusively by the Spatial Data Manager (SDM). The Mediator's architecture is also shown in Figure 2. Mediator uses M-SDM module for communicating with SDM. Its components cannot be treated as independent, because some of them are set of classes and methods for performing some of the functions of Mediator. ObjectManager is another Mediator module, which processes users run-time demands expressed over the OO schema, and maps them into corresponding database statements. In doing this, Mediator uses the definition of the application stored in MetaDataRepository and generates appropriate database statements. The component that generates

SQL statements and communicates with relational database is SQLGenerator. Mediator uses MDR to instantiate objects from relations. When the user updates some object, Mediator determines which tables and tuples need to be updated, and generates appropriate database statements. MDR also supports schema evolution. OIDGenerator is a component, which works with object identifiers. The Mediator ObjTab table contains information about all objects in the memory. Users can easily change schemas without even having to change the underlying code manually. Mediator provides data persistence, version control and schema evolution, as well as the run-time application support by interpreting user demands, invoking appropriate methods and generating database operations

As it can be seen on figure 2, the user could use only “passive” features of GinisNT environment using the users interface services. If the user uses active functionality, the possible effect of this action could be seen through GinisNT user interface. The active behavior is incorporated in Active Mediator using ECA (Event-Condition-Action) rules, which are part of database schema [16]. ECA rules are used to capture active capability of the system. The rules are defined at the same level as classes of application. This approach provides flexible control in different kinds of operation (low-level, methods, programs). The Active Mediator have to be capable for automatically and effective detecting primitive and complex events. After event’s detection, the Active Mediator performs rules associated with happened events.

The basic components of the Rule Manager layer is presented on figure 2 also. The Rule Manager provides all necessary tools for definition of active behavior. The RuleDef is a tool for rule definition, examination and connection with events, which are defined with EventDef component. EventDef is checking syntax and semantics of rules (using Analyzator functions) and suggest to the user how to correct the mistake. In this phase the EventDef component may define only primitive events but not complex events or transaction. After syntax and semantics checking, The Compiler translates ECA constructions to object definitions in GinisNT environment.

Event Detector controls all relevant events defined by user. After detection, the Event Detector sent message to the Rule Executor. This component read rules definition from the Rule Base. Every rule with signalized events is triggered and it could be activated. After that, rule condition is tested, and if the conditions are satisfied, the action can be executed. The Event Detector in this phase provides detecting only for primitive events.

The active features are integrated in existing object-oriented model. At this stage, rules and events is modeled as classes and implemented in C++. Using GinisNT services, the definition of events and rules are kept in GinisNT database as objects.

RULE <rule_name> <rule_id>
ON_EVENT <event_id>
IF <condition>
THEN <action>

Fig. 3 ECA Rules in ActiveMediator.

Every rule (figure 3) has a name for identification. ON_EVENT clause indicates event, which may activate the rule. One event could be connected to the lot of rules. The Active Mediator Rule Definition Language provides constructions for event’s definition. Condition in IF clause determines if the happened event triggered the rule. If the condition is satisfied, the rule is triggered and the action given in THEN clause is performed.

5. STOMM

In order to provide unified and integrated modeling, processing and management of spatial, temporal and thematic components of geographic information, STOMM, a general, spatio-temporal object modeling and management system has been developed on top of GinisNT [20]. It incorporates STOM data model and spatio-temporal object database kernel and integrates full spectrum of appropriate components for specialized temporal GIS application functionality built on top of STOMM Kernel. The STOMM presents a generic, temporal GIS application framework for development of specific spatio-temporal database applications.

An extensible database kernel - STOMM Kernel (figure 4) with classes and methods for representation and management of spatio-temporal data objects, spatio-thematic data updating component (Update Manager), mediation component for mapping between object-oriented application and (object-) relational database model (Mediator) and spatio-temporal indexing components. The STOM data model and its STOMM Kernel implementation can be extended by additional classes, attributes and methods, which facilitates the customization of the STOMM system to serve any particular spatio-temporal database application.

A collection of components for advances spatio-thematic-temporal data management such as: spatio-temporal query processing (STOQL), spatio-temporal object visualization and animation manager (STOVAM), spatio-temporal data acquisition, conversion and exchange, spatio-temporal analysis, reasoning and data mining.

A collection of graphical user interface components which can be incorporated in temporal GIS application user interface for accessing functionality of advanced spatio-temporal data management components for query processing, visualization, data acquisition, analysis and reasoning.

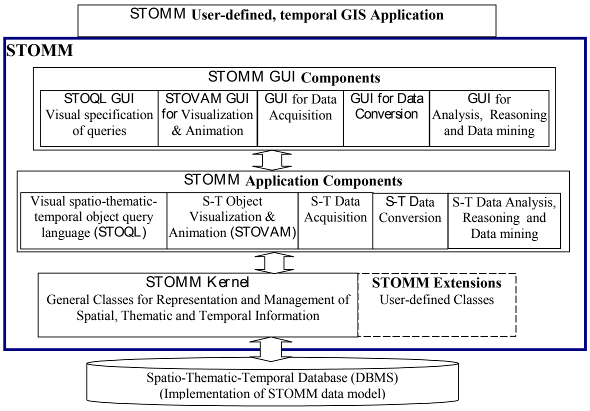

Fig. 4 STOMM architecture.

6. CONCLUSION

In this paper, we address the problem of turning a object-oriented spatial database into an active, temporal database management system without changing the underlying system.

The GinisNT system has layer architecture. Every component is developing on-top passive spatial database system. It was used GinisNT OO spatial database, which is based on RDBMS. The wrapper-mediator approach divides the functionality of a data integration system into two kinds of subsystems. The wrappers provide access to the data in the data sources using a common data model. The mediators provide coherent views of the data in the repositories by performing semantic reconciliation of the common data model data representations provided by the wrappers. The ActiveMediator, as an active layer on top GinisNT, will provide effective use of data in spatial database. Also, the Active Mediator will provide more influence of those data.

To provide support for temporal GIS application design and implementation, a suite of spatio-temporal object modeling and management technologies and tools has been developed constituting spatio-temporal object modeling and management system. The openness and scalability of developed frameworks enable their further improvement and refinement to satisfy the needs of specific, temporal GIS applications.

The advantages of this approach are numerous:

- Application framework provides object-oriented application development methodology and run-time application support.
- Mediator encapsulates access to databases for MS Visual C/C++ classes and it provides higher productivity, quality and reusability of classes.
- Mediator reduces changes to models of application. The changes that user makes in the class hierarchy do not influence the database; those changes are only reflected in the metadata repository.
- High quality of developed applications. GinisNT provides all classes for realization of GIS applications. The developers can just add data attributes to these classes when developing a simple application.
- Transparency of the underlying database and ability to add active and temporal capability without changing the user applications, retain relational DBMS's underlying functionality.
- Updating changed with current information achieving minimal data redundancy, evidencing changes and maintaining references between different data versions.
- Development and integration of components containing specific methods for spatio-temporal analysis and reasoning, which highly depend on specific temporal GIS application.
- The environment is simple and runs on cheap hardware platforms.

There are several open directions for future research. We firstly must extend our spatio-temporal data model to appropriately represent, store and retrieve continuously changed spatial objects, i.e. moving objects. Such topic has gained much more importance recently, with the advance of wireless communications and wireless Internet, mobile positioning using GPS- or GSM-based technology, and according to it, the great interest and applicability of location-based services. At this stage our model supports discrete events and changes in a stepwise manner.

Also, our future research activities are refered to defining an mediator-based architecture for integration of distributed and heterogeneous GIS data sources and adding the integration technology to the existing framework. We examine a research whose final goal is to make disparate data sources work together.

REFERENCES

[1] ACT-NET Consortium, The Active Database Management System Manifesto: A Rulebase of ADBMS Features, SIGMOD Record, Vol. 25, No.3 pp.40-49, September 1996.

[2] Al-Taha, K K, Snodgrass, R T, Soo, M D: Bibliography on spatiotemporal databases, International Journal on Geographic Information Systems, Vol. 8, No. 1, 1994, pp. 95-103.

[3] Boehlen, M H, Jensen, C S, Skjellaug, B: Spatio-Temporal Database Support for Legacy Applications, In: Proc. 1998 ACM Symposium on Applied Computing, Atlanta, Georgia, 1998.

[4] Collet, C, Coupaye, T: Composite Events in NAOS, Proc. of the 7th Int. Conf. on Database and Expert Systems Applications (DEXA'96), Zurich, Switzerland, September 9-13, 1996.

[5] Fahl, G, Risch, T, Sköld, M: AMOS - An Architecture for Active Mediators. Technical Report LiTH-IDA-R-93-13, Dep. of Computer and Information Science, Linköping University, Sweden. http://www.ida.liu.se/publications/cgi-bin/tr-fetch.pl?r-93-13+ps

[6] Gatziu, S, Dittrich, K R: SAMOS: An Active, Object-Oriented Database System, Buleltin of the IEEE TC on Data Engineering 15, pp.1-4, 1992.

[7] Guenther, O, Lamberts, J: Object-oriented Techniques for the Management of Geographic and Environmental Data, The Computer Journal, Vol. 37, No. 1, 1994, pp. 16-25.

[8] Gueting, R H: An Introduction to Spatial database Systems, VLDB Journal, Vol. 3, No. 4, 1994, pp. 357-399

[9] Langran, G: Time in Geographic Information Systems, Taylor & Francis, 1992.

[10] Li, L, Chakravarthy, S: An Agent-Based Approach to Extending the Native Active Capability of Relational Database Systems, In Proceedings of the 15th International Conference on Data Engineering, pages 384-391,. IEEE Computer Society Press, 1999.

[11] Mitrovic, D, Mitrovic, A, Djordjevic-Kajan, S: A scalable, object-oriented GIS Framework, In: Proc. ISPRS Int.Workshop Hardware and Software Aspects of GIS, Milan, Italy, 1996, pp. 20-21.

[12] Pavlopoulos, A, Theodoulidis, B: Review of SpatioTemporal Data Models. Technical Report TR-98-3, TimeLab, Dept of Computation, UMIST, Manchester, Great Britain,1998.

[13] Roth, M T, Schwarz, P: Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, In: Proceedings of the 23rd VLDB Conference, Athens, Greece, 1997, pp.226-275.

[14] Snodgrass, R T: Temporal Databases, in: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, A. U. Frank, I. Campari, U. Formentini (eds.), LNCS. 629, 1992, pp. 22-65.

[15] Snodgrass, R T: Temporal Object-Oriented Databases: A Critical Comparison. Modern Database Systems, W. Kim (ed.), Addison-Wesley/ACM Press, 1995, pp. 386-408.

[16] Stoimenov, L, Djordjevic-Kajan, S, Mitrovic, A: Active component of the GinisNT Mediator, In: Proc. YUINFO'98, Kopaonik, 1998, pp. 15-18. (in serbian)

[17] Stoimenov, L, Djordjevic-Kajan, S, Stojanovic, D: Integration of GIS Data Sources over the Internet Using Mediator and Wrapper Technology, In: Proc. MELECON 2000, 10th Mediterranean Electrotechnical Conf., Cyprus, May 29-31 2000, Proc. Vol. 1, pp. 334-336.

[18] 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 Science, Vol. 41, No. 2, 1999, pp.59-68.

[19] Stoimenov, L., Mitrovic, A, Djordjevic-Kajan S: An Object-Relational Environment For Developing Object-Oriented GIS Applications Based On Mediator, Presented on 16th IASTED International Conference on Applied Informatics, Garmisch- Partenkirchen, Germany, 23-25. February 1998.

[20] Stojanovic, D, Djordjevic-Kajan, S, Stojanovic, Z: Modeling and Management of Spatio-Temporal Objects Within Temporal GIS Application Framework. In: Proc. International Database Engineering and Applications Symposium, Grenoble, France, July 2001, pp. 249-254.

[21] Tansel, A U, Clifford, J, Gadia, S, Jajodia, S, Segev, A, Snodgrass, R: Temporal Databases: Theory, Design and Implementation, Database Systems and Appliactions series, The Benjamin/Cummings Publishing Company Inc., Redwood City, CA, 1993.

[22] The TSQL2 Temporal Query Language, R. T. Snodgrass (ed.), Kluwer Academic Publishers, 1995.

[23] Vijbrief, T, van Oosterom, P: The GEO++ System: An Extensible GIS, In: Proc. 3rd International Simposium on Spatial Data Handling, 1992, pp. 40-50.

[24] Wiederhold G: Weaving Data into Information, Database Programming & Design, Vol. 11, No. 9, Sept.1998, pp.22-29.

[25] Wiederhold, G: Mediators in the Architecture of Future Information Systems, IEEE Computer, Vol. 25, No. 3, March 1992, pp. 38-49.

BIOGRAPHY

Leonid STOIMENOV was born on 12.07.1967. In 1997 he graduated (MSc.) with distinction at the department of Computer Science of the Faculty of Electronic Engineering at University in Niš. He defended his PhD. in the field of GIS applications interoperability based on mediators and ontologies. Since 1992 he is working as a tutor with the Department of Computer Science. His scientific research is focusing on GIS, databases, mediators and ontologies. He also investigates questions related with the semantic interoperability of heterogeneous information sources.

Dragan STOJANOVIĆ was born on 1968. In 1998 he graduated (MSc.) with distinction at the department of Computer Science of the Faculty of Electronic Engineering at University in Niš. His scientific research is focusing on GIS, spatio-temporal databases, spatio-temporal data models and location based services.