reviewed paper

Building Smart Applications for Smart Cities – IGIS-based Architectural Framework

Alexander Vodyaho, Nataly Zhukova

(Dr. Alexander Vodyaho, LETI, 5, Popova str., St. Petersburg, 197376, Russia, aivodyaho@mail.ru)
(Dr. Natalia Zhukova, SPIIRAS, 39, 14th Line, V.O., St. Petersburg, 199178, Russia, gna@oogis.ru)

1 ABSTRACT

To solve different kinds of complicated problems which arise in context of intensive development of modern cities a great number of various applications are constantly being developed. The most part of these applications are based on processing big volumes of heterogeneous data gathered from different types of available sources in real time. In the report an architectural framework oriented on building applications for smart cities in shortest time and with minimum spent of resources is suggested. The framework is based on intelligent geo information technologies and includes architectural and technological solutions along with many different computational libraries for building intelligent adaptive applications. Special attention is paid to information and knowledge organization. Different aspects of use of ontologies in the framework is discussed. Main directions of further development of proposed approach are defined.

2 INTRODUCTION

Software applications build for needs of cities have almost always been one of the main consumers of new solutions developed in the sphere of information technologies (IT). Moreover they often define direction for development of technologies and force the IT to onrush continuously to meet constantly increasing requirements.

Unfortunately, the current state of IT as a whole is much more poor than it was several years ago. Following negative tendencies are observed in the IT sphere nowadays.

(1) The sphere of IT gradually loses the status of the sphere in which business is ready to make essential long-term investments without taking into account short term expenses. Today for the majority of enterprises the IT is one of many services which should be estimated in the terms of ROI and moreover, a number of investors want to return earlier invested funds.

(2) Complexity of the developed information systems is permanently increases. One can say that the Moore’s law can be applied to information systems.

(3) The level of qualification of IT specialists is gradually decreasing. During last years popularity of technical education significantly decreased. That leds to reduction of the number of highly qualified specialists. IT companies mostly prefer to employ rather cheap foreign programmers.

Along with negative tendencies there are several positive trends caused by two main factors. The first factor is that during the period of information technologies active development many architectural and technological solutions were proposed, implemented and approved. The second factor is that high performance tools, including tools that use artificial intelligence technologies, were build and they have become an essential part of the advanced information systems. The bright example of such systems is intelligent geoinformation systems (IGIS) [1]. Means and tools integrated in IGIS include inference engines, expert systems, libraries of various intelligent algorithms, instruments for data, information and knowledge management.

As a response to this situation industrial approach to software application development and support has been worked out. The main features of the developed approach are following.

(1) Wide use of best practices that are represented in the form of the frameworks.

(2) Use knowledge-oriented technologies for software development.

(3) Assembling information systems using large-scale program modules.

(4) Use various agile decisions and practices [2]. Agile decisions are decisions that can be easy adapted to the specific conditions (contexts). Agility can be presented in different forms, in particular:

(4a) agile software development – the process of software development that allows working with constantly changing requirements;

(4b) agile architecture – architecture of software that allows develop architecture of systems oriented on solving specific problems of the subject domains;

(4c) agile algorithms – algorithms that are context sensitive and self adjustable.

Nowadays the industrial approach is already successfully applied for building applications in different subject domains [3]. It has become quite obvious that technologies for constructing, developing and supporting agile applications will be further developed. So it is time for specialists in information and communication technologies for smart cities to look towards the industrial approach for building applications and to adapt the proposed solutions to their needs.

Information and communication technologies in modern smart cities are of primary importance as they form the backbone for all integration processes that take place inside and between such spheres as social, economic, industrial, environmental and etc.

The following consequences of integration processes influence directly on requirements imposed to IT solutions:

(1) many software applications were integrated using various technologies and formed a net of interconnected applications that are poorly managed and supported;

(2) in the integrated domains dynamics and complexity of both internal and external processes exponentially increases, furthermore processes as a rule can not be formalized and are unpredictable;

(3) established interbranch relations provide possibility to solve multidisciplinary problems that are much more complicated; besides, experts in one subject domain are forced to solve specialized tasks from the subject domains that are not in the area of their competence.

To meet the requirements of software applications for smart cities, that are capable to support integration processes, IT solutions for smart applications must be developed according to the principles defined below.

(1) Smart applications must be knowledge-centric applications in order to provide possibility for a user to work effectively with them.

(2) Construction of applications must be oriented on integration of technologies.

(3) It is necessary to use a unified high level base platform that can be adapted to concrete subject domain. Using base platform for building applications provides a unified information space, mechanisms of platform adaptation will allow to develop applications in conditions of limited resources.

Taking into account the current state of IT sphere and smart cities, the developed framework must be a knowledge-centric agile framework. Also an unified approach for creation and support domain-oriented applications using the framework and the base platform must be worked out.

In the paper the knowledge based domain-oriented architectural framework for smart cities applications is proposed. In the following section existing frameworks and possibilities of their application are discussed. In the fourth section an architectural approach for constructing domain-oriented applications with the help of the developed framework and the base platform is considered. In the fifths section questions of knowledge management and usage in the framework and the end applications are discussed. Implementation of the framework is described in the last section.

3 ARCHITECTURAL FRAMEWORKS – COMMON SOLUTIONS

In terms of an architectural framework an approach to development software applications is considered [4]. Architectural frameworks are used, first of all, for creation architectural descriptions of end software products, families and lines of products and for developing architectural modeling tools both for one or for groups of organizations. The architectural description is an artifact or set of artifacts that describe the architecture of the developed application. An architecture is an abstraction, that includes concepts and properties of an application.

By today many architectural frameworks have been developed. The most well known and widely applied are the following frameworks: Zachman Information Systems Architecture Framework [5], UK Ministry of Defense Architecture Framework [6], The Open Group's Architecture Framework (TOGAF) [7], Kruchten's "4+1" view model [8], Siemens' 4 views method [9], Reference Model for Open Distributed Processing (RM-ODP) [10] and Generalized Enterprise Reference Architecture (GERA) [11].

The term architectural framework is closely connected with such terms as platform, paradigm and pattern. A paradigm describes the concept, basic states and terms used for developing applications. A platform is an implementation of the framework that provides building blocks for end applications and tools for applications development. Following types of platforms are usually considered: technological, integration and domain-oriented platforms [3]. Domain-oriented platforms are platforms aimed for creating applications for one or several related subject domains or for solving a certain group of tasks. Patterns are complete fragments of program code that can be used many times without modifications and improvements.

Along with typical frameworks listed above a wide range of specialized platforms have been developed. From the point of view of problems that are solved in modern cities it is interesting to consider two of them. The first framework is an intelligent geo information (IGIS) framework described in [1]. This framework provides scientific and technological solutions for a wide range of highly demanded tasks such as monitoring and decision making support at the levels of objects and situations as well as tools and means of artificial intelligence. Along with that IGIS framework allows to organize a convenient workspace for end users. The second essential framework is a framework for data processing and analyses (DPA) [12]. This framework allows processing various types of data including complicated time series of measurements. Results of measurements are continuously gathered using different kinds of specialized instruments that are by now installed almost everywhere both in urban and environmental areas. For the IGIS and the DPA frameworks corresponding platforms have been build. Below a brief description of each platform is given.

IGIS platform. The IGIS platform incorporates the following list of basic components: an inference machine and expert system, a knowledge base system (ontology), visual environment for developing classes and objects of subject domain, visual environment for developing models (script) of the objects behavior in GIS, system for scenario implementation in real time or/and user-defined arbitrary scale with visual display of symbols or images on the background of electronic maps, a decision making support system that provides recommendations during the scenarios playing.

In the IGIS platform an expert system and ontologies are considered as a system of artificial intelligence. Expert systems are used for solving two main tasks: assisting a decision-makers and managing various processes working under control of a scenario. A scenario is the selfsame algorithm with a capacity for parallel execution of some of its branches, it has ontological representation and can be interpreted by inference machine.

DPA platform. The DPA platform contains following basic components: an ontology of measurements and ontology of methods and means of data processing and analysis, means and tools for knowledge extraction from historical data, means and tools for working with business processes, mathematical and empirical methods and algorithms for processing and analysis of large volumes of data, methods and algorithms of preliminary data processing, algorithms of prospecting data analysis, a set of visual tools that can be used for graphical representation and modification of initial data streams, as well as time series and single measurements and the results of their processing.

Special attention in the DPA platform is paid to two moments. The first moment is development of agile algorithms for multidimensional measurements processing received from natural and technical objects in real time and in the delayed mode. The second moment is wide usage of implemented intelligent data processing and analysis technologies, based on both original and commonly used algorithms and patterns.

The IGIS and DPA frameworks are considered to be the core elements of the framework for building applications for smart cities. The specialized frameworks developed for applied domains, for example for the industrial domain [3], can be considered as extensions and included into the described framework for smart cities.

4 DOMAIN-ORIENTED ARCHITECTURAL FRAMEWORK FOR SMART CITIES

The proposed architectural approach is focused on development of a framework for the domain of management and planning of modern cities economy oriented on solving applied problems in various spheres that are a part of cities economy. The framework is described according to the international standard ISO/IEC/IEEE 42010:2011 [13].

The framework is developed using the following basic solutions for constructing software applications (SA) for smart cities:

(1) for constructing architecture of the SA for smart cities Model-Driven Engineering (MDE) methodology [14], based on hierarchy (stack) of architectural models is used. Models are containers for architectural descriptions. The stack contains models of four different levels of architectural descriptions;

(2) SA for smart cities must be implemented according to their architectural description using base program platform that contains program components, modules, common means and tools;

(3) in the process of SA construction available architectural knowledge, including knowledge about best practices is to be obligatory used;

(4) constructing SA includes development of the knowledge base required for solving end user problems and for supporting applications.

To make the first two moments of the listed solutions realizable it is necessary to develop a base program platform and a stack of architectural models, adapted for subject domain of cities economy. The architectural models are build according to the object oriented (OO) models. Thus, it is supposed, that before each system is constructed, an OO model for the application according to [16] is created.

The two last points make the developed applications and the processes of their constructing knowledge oriented. For dealing with knowledge a system of ontologies and solutions for their transformation were developed. The description of the proposed system of ontologies is presented in the corresponding section.

The framework allows to create architectural descriptions for the end applications on the base of their object-oriented descriptions, that can be implemented using the base program platform. The relations between the framework, the object oriented model of an application, the architectural model and the base platform are shown in Fig. 1.

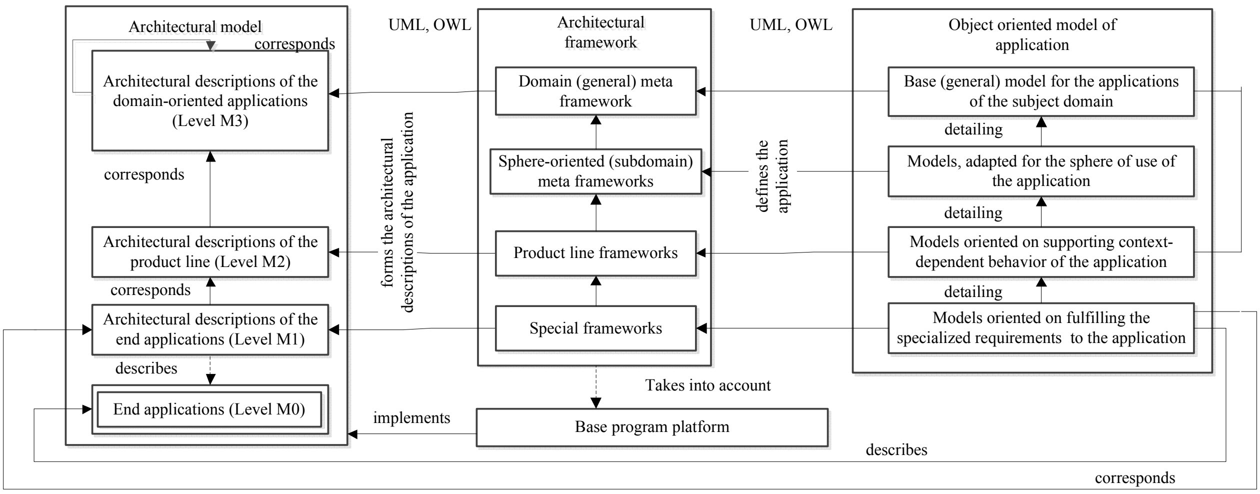

Fig. 1: Main elements used for constructing applications with the help of the domain-oriented architectural framework

Object-oriented model of an application. An object oriented model of an application includes the description of the subject domain, where the application is going to be applied, the tasks that are supposed to be solved and the requirements to the application. The model contains descriptions at different levels of abstraction. The description, that corresponds to the lowest level, is the description of the end application. The description of the highest level is the general description of all applications for the defined subject domain or subdomain. At the middle levels peculiar features of the application that are defined by the possible spheres and contexts of its usage are described.

Architectural model. The stack of the architectural models that are described in a general form in the MDE methodology are adapted for constructing software applications for smart cities. The lowest level of the models hierarchy (level M0) corresponds to the architectural descriptions of the constructed applications. The descriptions reflect requirements imposed to the applications taking into account the groups of users and their specific needs. The level M1 is the level at which descriptions of the applications architecture are defined according to different contexts in which the applications are supposed to be applied. At the level M2 descriptions of the applications adapted for a certain group of tasks or one of the spheres of the subject domain, where the application will be used are constructed. The level M3 is the level, at which the general descriptions of the applications for the subject domain of cities economy or its subdomains are provided.

All models used, at levels M1, M2 and M3, are abstract models. The result architectural description that is created using the proposed hierarchy of models contains a complete architectural description of the developed application including the description of the stakeholders and their concerns, all architectural viewpoints, types of models for each of the viewpoints and architectural rationales [13]. The model types define the language and main notions such as modeling techniques that are to be used for describing view points.

For the stack of the architectural models operations for models transformation based on the developed technique are provided [14]. Application of transformation operations allows to create a new model on the base of one or several existing models. A transformed model and a result model can belong to the same or to different levels. In the first case horizontal transformations are executed and in the second – vertical transformations. Results of horizontal transformations are models of a higher level of abstraction than the transformed models.

Links between different models at one level can be established with the help of binding operations. Almost all elements of models including classes, objects and their properties as well as relations between them can be linked.

To support both transformation and binding operations corresponding set of transformation and binding models according to [14] were developed. The models are represented in a form of patterns and rules.

Architectural framework. The architectural framework is developed according to [13] and adapted to current needs of modern cities. The framework is organized as a hierarchy of frameworks that has the following structure. At the highest level a general domain-oriented framework (DF) is located. The DF is a problem-oriented meta framework that does not apply any restrictions on the architectural solutions of the developed software applications. The domain framework is used for building sphere-oriented meta frameworks (SF) that are aimed to solve problems of one or several related subdomains. The SF form the base for special frameworks. Two types of special frameworks are used – the frameworks for constructing lines of products (LF) and for constructing end software applications (F). The architecture of each end application can correspond only to one F-framework. LF-frameworks and F-frameworks can be build on the base of several SF-frameworks inheriting different concerns, viewpoints and etc. In case a product line is developed F-frameworks are always based on LF-frameworks. Each LF-framework can be used to build multiple F-frameworks, but one F-framework corresponds only to one FL-framework.

The distinguishing features of the DF-framework is that the constructed architectural descriptions allow building knowledge-centric software applications where the applied problems of cities economy are solved using data fusion technologies along with the commonly used technologies implemented in earlier developed applications for the considered subject domain. The data fusion technologies are based on JDL model [15]. Two implementation of JDL model are supported – implementation, oriented on extraction of information and knowledge from initial data that is gathered from measurement instruments or received from data centers developed for the DPA platform, and implementation oriented on solving complicated applied tasks including decision making support provided by the IGIS platform.

Main advantage of the proposed DF-framework is that on its base software applications that provide principally new quality of data processing, information and knowledge about cities economy on the base of existing solutions implemented in IGIS and DPA platforms can be easily constructed.

The subdomain meta frameworks provide means and tools for defining following elements of an architectural description:

- main types of stakeholders, including end-users, operators, acquirers, owners, developers, builders, maintainers and software architects;
- the system of typical interests, including serviceability, cost, maintainability, stability, analyzability, changeability, testability, dependability, modularity, distribution and concurrency;
- viewpoints, including general, algorithmic, capability, system viewpoints, data, information and knowledge viewpoint, object, services, project and standards viewpoints.

Between the enumerated elements main relations such as relations between interests and stakeholders, stakeholders and architectural viewpoints, types of models for each of the viewpoints are considered.

ISBN: 978-3-9503110-6-8 (CD-ROM); ISBN: 978-3-9503110-7-5 (Print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI

The LF-frameworks are aimed for building a limited set of the architectural descriptions elements that are formed using the information about the supposed users of the produced software products.

At the level of F-frameworks architectural styles that are supposed to be used for constructing end applications are defined. The following common architectural styles and corresponding architectures are considered: object (component) oriented architectures [16], service-oriented architectures [17], multiagent architectures [18], combined (mixed) architectures. In addition service-agent-service (SAS) architecture was developed that refers to combined architectures. The SAS architecture fits best in most cases for building end applications for smart cities.

Base program platform. The base program platform is a software application, that is used for implementation of the developed architectural descriptions. The platform provides core elements, that are required for building end applications and a wide range of various mathematical libraries, as well as extended means and tools of artificial intelligence inherited from IGIS and DPA platforms and from other integrated platforms.

5 A SYSTEM OF ONTOLOGIES FOR THE ARCHITECTURAL FRAMEWORK FOR SMART CITIES

Ontologies in the SA developed for smart cities are widely used at all stages of the applications life cycle: at stages of SA construction, development, functioning and support. At the stage of application construction experience acquired during the previous developments is actively used. Experience is represented in the form of knowledge using standard formats, in particular, the OWL format [19].

Knowledge-based descriptions of the developed SA supplement the architectural descriptions, build according to the MDE methodology. Thus, the result architectural descriptions of the end applications are sets of descriptions in UML and OWL formats. The OWL descriptions provide information about the main tasks and the developed architectural solutions as well as the descriptions of possible ways of the application further development and modification. The UML descriptions contain detailed information about the solutions of the tasks, in particular the structures of the SA. It is important to note that the two considered descriptions can be transformed one to the other. The problems of the direct and indirect transformations of the descriptions have been worked at for a long time. As a result such means as Metadata Interchange (XMI), Ontology Definition Metamodel (ODM), UML Profile (OUP), that allow to create ontologies using UML descriptions, are available now [20].

The structure of the developed system of ontologies is defined by the hierarchy of the architectural models and frameworks discussed in the section 4. The ontologies provide descriptions of the applications life cycles at the domain level, at the subdomain level and at the level of applied tasks, that are used for describing architecture at the third, the second and the first levels of the architectural model correspondingly.

Along with the main task of building knowledge based architectural descriptions, one can use ontologies for solving following tasks:

(1) ontologies that describe the subject domain are used by analysts for defining problems of the SA construction;

(2) a data base structure can be created from the ontological descriptions of the subject domain;

(3) an architectural descriptions of the user interfaces can be created by converting the information that contains in the ontologies;

(4) in multiple cases architectural descriptions of the constructed applications or their separate components can be build by converting information from ontologies;

(5) using ontological descriptions a system of rules can be defined, that describes behavior of the objects of the subject domain and can be interpreted and executed using standard tools;

(6) ontologies are used as a dictionary that provides descriptions of the main definitions of the subject domain to all stakeholders.

When SA are constructed, a set of ontologies and ways of their usage at the stage of the applications functioning is defined. The ontologies are build partly at the stage of SA construction and partly at the stage of SA development.

At the stage of applications functioning ontologies are used for solving two main groups of problems. To the first group refer problems of data, information and knowledge transformation, to the second – problems of management of the transformation operations.

For SA support and modification special ontologies are developed, that are a kind of slices of ontologies, build at the stage of the SA construction. These ontologies contain limited amount of information that allows understand in short time the general architecture of the applications, estimate their current state and make well-founded decisions about the further changes of the applications structure and/or business logic.

At all stages of the SA lifecycles the following techniques for working with knowledge presented in the form of ontologies are used: i) acquiring knowledge from data (data mining); ii) acquiring knowledge from log files about executed business processes (process mining); iii) enlarging ontologies by adding new knowledge provided by experts; iv) performing logical deduction; v) transformation of knowledge presented in the form of ontologies (aligning, merging, building profiles); vi) representation of knowledge in the form of production rules; vii) building UML and ER descriptions from ontologies.

In the architectural framework the process of building systems of ontologies for end applications is organized on the base of the developed ontological model. The levels of the model and the artifacts of the levels are shown in Table 1.

Level of the ontological model	**Main artifacts of the level**
Domain / subdomain level	Domain / subdomain ontologies
Level of product line	Ontologies that contain information about the solved tasks
Level of the end applications	Ontologies required for the applications functioning, support and modification

Table 1: Artifacts of the ontological model at different levels of application construction.

An ontological model has three levels, that correspond to the SA construction at the domain / subdomain levels, at the level of products line and at the level of end applications. At the third level ontologies for the subject domain or its subdomains are developed. Ontologies for product lines are domain ontologies that additionally contain information about the problems, that are supposed to be solved using developed applications. The main goal of the first level is to build ontologies that are to be used during the stage of the SA functioning as well for the needs of SA support and modification.

Ontologies of the domain and subdomain levels are commonly developed by the experts of the corresponding subject domains. In the processes of the ontologies development in most cases analysts and knowledge engineers are involved as well. Ontologies for product lines and ontologies, used for the software application support and modification, are build by analysts and IT specialists on the base of domain ontologies.

The list of ontologies, applied at the stage of the SA functioning, includes ontologies for solving both specialized and common tasks. Ontologies, oriented on solving specialized tasks, are build by experts The ontologies for solving common tasks are constructed on the base of the higher level ontologies, that are provided by the corresponding domains. For example, for the domain of data processing and analyses a set of developed ontologies contains general formalized descriptions of data of different types and results of its processing, of available algorithms, methods, means and tools for data processing and rules for their application, of the processes that can be used for solving various problems, of the program modules that can be included in the applications, possible providers of the modules and etc.

6 IMPLEMENTATION OF THE ARCHITECTURAL FRAMEWORK FOR SMART CITIES

Proposed solutions for implementing end SA using the architectural framework are agile solutions [2], that are developed within the industrial approach. The solutions are based on the following main principles:

(1) multiple use of knowledge is preferable in comparison with multiple use of program code;

(2) the developed applications have to be based on the integrated set of the existing ready to use technological solutions. Much less desirable but still admissible is to integrate existing solutions at the level of program components and modules. New modules are developed only if there are no other alternatives;

ISBN: 978-3-9503110-6-8 (CD-ROM); ISBN: 978-3-9503110-7-5 (Print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI

(3) the implementation of the new program modules must be organized from the point of view of possible subsequent use of the modules. The resources spent on the modules that can be multiply used are significantly more justified than the resources spent on the development of the unreusable modules;

(4) solutions for application implementation have to provide opportunities to modify developed applications structures and business logic on the fly by both IT specialist and applications themselves. For this the description of the applications must be represented using standard interpretable formats. For describing applications in most cases ontologies are used, as, on one hand, they can be considered as means of standardization, and, on the other hand, as means capable to deal with knowledge.

Implementation of SA based on the agile solutions that are supported by the architectural framework are based on the following notions: A - system (agile system), A - process (agile process) and A- application (agile application). A – system is a set of integrated A - applications. A - process is the knowledge-based process of constructing A - applications. A - application is a self-adjustable application that contains a knowledge base and means of artificial intelligence. Two types of A-applications are allocated: static and dynamic A-applications. Both static and dynamic A-applications are constructed using A-processes. Main difference between these types of applications is that static applications don't use ontologies at the stage of their functioning meanwhile in the dynamic applications ontologies are often considered as a core element. Agile applications have a hierarchical structure. At their top level two subsystems are located, that are represented in the form of containers for modules. The first container (B - container) provides implementation of business functions which define functionality of the applications. The second container (A - container) implements functions of management and control of the B - container architecture. A and B containers in separate cases can be considered as modules.

Five types of modules are used in A – applications: A-modules, B-modules, AB-modules, R-modules and V-modules. A-modules are modules with the dynamic architecture that support A-interfaces which provide access to the functions that assure module agility. These modules are executed under control of the A-container. B-modules support B-interfaces and define system functionality. These modules have static architecture and are managed by a B-container. B-modules are able to provide information about their state. AB - modules support both types of the interfaces and interact with the A and the B containers. This type of modules is used most frequently in the applications. R-modules are aimed for working with various types of repositories were initial data as well as meta data is stored. The V-modules contain the implementation of visualization functions and support interaction of an application with an end user.

The considered agile solutions don't impose any restrictions on implementation of interfaces, or on implementation of the containers. Interfaces are described in terms of applied technologies. The containers implement functions of modules integration, their management and control during their life cycle, and, respectively, the containers have to support the chosen technologies.

The concrete technologies for each end SA are selected within F-framework used for constructing end applications.

Usage of the object-oriented approach [17] for building applications supposes that B-containers are business objects (B-objects) described using the high level language or represented in the form of components (.net, CORBA, EJB). A-containers are sets of A-objects that are responsible for managing B-objects and gathering data about the applications functioning. Data processing and analyses is organized in a special subsystem of an A-container or a separate external subsystem.

Service-oriented approach (SOA) [18] can be used for implementation both static and dynamic architectures. Implementation of static SOA implies that a set of service modules (S-modules), which include business services corresponding to B-objects, and agile services corresponding to A-objects are developed. For implementation of SA, business logic on the base of dynamic SOA services can be organized with the help of semantic web services.

Usage of agent technologies in end applications assumes that A-subsystem is a set of agents, that includes personal assistances, business agents, agent managers, agents for agility support, agents for access to metadata and data, agents for support of B2B interactions [19].

Implementation of the enumerated technologies is supported by the base program platform.

7 CONCLUSION

The paper aims to find an approach for building SA oriented on solving different problems in various spheres of modern cities economy. The approach must be able to deal with challenges of modern smart cities in conditions of the quite poor state of information technologies.

In the paper the architectural framework for building SA for smart cities is proposed. Distinguishing features of the framework can be summarized in the following points:

a) the framework is knowledge-based. It means that knowledge is used at all stages of the SA lifecycle (construction, functioning, support and modification);

b) the framework allows to construct SA by means of integration of existing solutions at technological level and at the level of program modules;

c) solutions provided by the framework are agile:

- architectural description of a SA is built on the base of object oriented descriptions;
- architectural descriptions built with the help of the framework can be implemented using the base program platform;

d) the framework is developed according to the actual architectural and technological standards.

The proposed architectural framework described within the paper was developed as a generalization of the developed architectural, technological and program solutions, that were implemented as product lines and separate products and have been successfully used for a number of years already. The following software products defined the structure of the framework and its functions:

- the Ontomap series developed in the Research laboratory of object-oriented geo-information systems of St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences. The Ontomaps are used in Russian Navy for more than five years as the information base of Navy control systems. They are operated in Russian Navy fleet command centers, naval bases and combat information center of Coast Surveillance System;

- the series of software applications for telemetric information processing from space rockets developed in the Research Center of Saint-Petersburg Electrotechnical University. The total number of developed application is more than fifty. The applications are widely used for processing and analyses of initial data, calculation of flight, ballistic and navigation characteristics;
- a set of separate applications oriented on processing environmental parameters. For example, the Decision Making Support System for Arctic Exploration, Monitoring and Governance [21] was developed. The main goal of the system is to synthesize ocean/ice/atmosphere observations and model-based products for the purpose of fast access to the available information on the Arctic environment.

Future work needs to be carried out in the direction of further adaptation of the framework for the domain of cities economy. Adaptation assumes building ontologies on the base of available knowledge provided by existing software products and integration into the framework the earlier developed specialized solutions represented in the form of technologies, program components or modules. It is also reasonable to analyze the possibilities of adaptation and usage of new technologies developed for the spheres not related to the cities economy for processing, analyzes and management of data, information and knowledge of modern smart cities.

8 REFERENCES

1. Intelligent geographical information systems for sea situation monitoring [in Russian / Eds. Usupov R. & Popovich V. Nauka, St. Petersburg, 2013.
2. Hazzan O., Dubinsky Y. Agile Software Engineering. Springer, London: 2008.
3. Duffy D. Domain Architectures: Models and Architectures for UML Applications. John Wiley & Sons, New Jersey, 2004.
4. Kaisler S. Software Paradigms. John Wiley & Sons, New Jersey, 2005.
5. Zachman, J.A., A Framework for Information Systems Architecture, IBM Systems Journal, 26(3), 1987.
6. Ministry of Defence Architecture Framework (MODAF), http://www.modaf.org.uk/ [Electronic resource].
7. The Open Group Architecture Framework http://www.opengroup.org/togaf/ [Electronic resource].
8. Kruchten, P.B., “The ‘4+1’ View Model of Architecture”, IEEE Software, 12(6), 1995.
9. Hofmeister C., Nord R., Soni D. Applied Software Architecture, Reading, Addison-Wesley, MA, 1999.

10. Standard ISO/IEC 10746 Reference Model for Open Distributed Processing (RM-ODP) http://www.iso.org/iso/support/faqs.htm [Electronic resource]
11. Standard ISO 15704. Generalized Enterprise Reference Architecture http://www.iso.org/iso/support/faqs.htm [Electronic resource]
12. Vitol A., Deripaska A., Zhukova N. & Sokolov I. Technology of adaptive measurements processing. SPbSTU «LETI», Saint-Petersburg, 2012.
13. Standard ISO-42010. http://www.iso-architecture.org/ieee-1471/docs/ISO-IEC-IEEE-latest-draft-42010.pdf
14. Zivin B.E, Jouault J., Valduriez P. On the Need for Megamodels // Procs of the OOPSLA/GPCE: Best Practices for Model-Driven Software Development. Workshop, 2004.
15. Steinberg A., Bowman C. & White F. Revisions to the JDL Data Fusion Model. Sensor Fusion: Architectures, Algorithms, and Applications. Proceedings of the SPIE, vol. 3719, 1999.
16. Booch G., Object-Oriented Analysis and Design with Applications (2nd Edition), Addison Wesley, MA, 1993.
17. Erl T. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, N.Y, 2005.
18. Walton, C. Agency and the Semantic Web: Oxford University Press Inc. N. Y., 2007.
19. Calero C., Ruiz F., Piattini M. Ontologies for Software Engineering and Software Technology. Springer, Berlin, 2006.
20. Gasevic D, Djuric D., Devedzic V. Model Driven Architecture and Ontology Development, Springer, Berlin, 2006.
21. Zhukova N., Smirnova O. Atmosphere and ocean data processing in decision making support system for Arctic exploration // The 6th International Workshop on Information Fusion and Geographic Information Systems: Environmental and Urban Challenges (IF&GIS' 2013), St. Petersburg, Russia, May 12-15, 2013.