# OPTIMIZATION OF E-SERVICE DEVELOPMENT FOR LATVIAN PUBLIC SECTOR

**Pavel Semenchuk**

*Riga Technical University, Applied Computer Science Department,*
*Meza str. 1/3, Riga, Latvia, Pavels.Semencuks@rtu.lv*

**Abstract**. The development of Latvian State E-service was started in January, 2006 (Integrated State Information System (ISIS, Latvian abbreviation IVIS)). The customer and provider of the system is Secretariat of Special Assignments Minister for Electronic Government Affairs (SSAMEGA)). Since the middle of 2007 the infrastructure is accessible for E-service designers and developers. ISIS is based on Service Oriented Architecture (SOA) principles and includes many individual innovations and modern solutions. There are more than 60 E-services, which were developed using ISIS infrastructure, already accessible for Latvian inhabitants on Latvian government E-service portal (www.latvija.lv). These E-services consist of many Web services which are developed by different organizations and published in special register. The number of E-services growths up very quick. To avoid duplication of Web services from different organizations new solution was worked out and soon will be implemented in ISIS. This and other ISIS innovations are described further in this paper.

**Keywords:** e-government, SOA, logic programming, deductive database.

## 1 Introduction

Development of technologies provides new and effective methods all the time which allow improving work of data procession as well as mutual exchange and integration. These are important processes which have to be considered in planning and development of State Electronic Administration because in work of state administration are used several substantial registers where data about objects and subjects existing in their contents are stored as well as about services which the state has to provide according to legislation. Electronic service development and state electronic government implementation is one of priorities almost in all the most developed countries of the European Union and the world, Latvia included. It is very important that e-government and integration process would occur not chaotically, but unitarily. Preceding decade could be considered as time, when primary data registers were created. The new decade, evidently, will be oriented on these registers integration, offered services electronization and their accessibility for inhabitants. All developed countries of the world are going this way now.

Before ISIS system creation the world and another Baltic state experience was taken into account. Major part of attention was fixed on Estonian X-Road [12] system. ISIS was created on X-Road successfully usage principle: defined centralized data exchange point, standardized data exchange formats and protocols. The main difference is that ISIS tries to use already existing state registers and information systems data exchange infrastructure, if it exists. At the moment X-Road integrates 150 data bases and information systems and is used widely, but ISIS usage at the moment is only for e-services functionality providing. Lithuanian integration project (e-Gate) [4] goes similarly as ISIS project. It was started in 2004. In 2008 is planned, that two pilot e-services will be created for e-Gate project.

The article’s author have participated in development (system analysis, design, coding, and integration) of ISIS system from the very beginning, and ensured successful defence of the system publically in Riga Technical University State Qualification Commission. It proceeds to be developed during development process of Doctorate work. By use of ISIS infrastructure were developed already more than 60 E-services for Latvian inhabitants.

This paper is organized as follows. Section 2 describes the ISIS architecture. Section 3 describes the principals of logic programming and deductive data bases. Section 4 describes the optimization solutions for ISIS architecture and E-service development. Conclusions state further directions of research.

## 2 Description of ISIS solution

ISIS ensures a possibility mutually integrate state and municipal information systems, which now or in the future will be providers or recipients of various E-services including numerous portals, from which E-services will be accessible for inhabitants. Not only state or municipal institutions will take part in the development of the E-services, but also commercial institutions, for example, banks, which will be necessary to involve for paid E-services realization.

ISIS solution is based on the following principles:

- The use of Service Oriented Architecture (SOA).
- The solution technologically supports the integration with numerous E-service providers.
- Identification standard, which provides world scale unique number assignment.
- The use of state level unified standards for E-service and integration projects development E-service – XML schemas development standards [11], IS service development standards [5], E-service development standards [3], SOA standards.
- Four centralized catalogues development – XML schemas catalogue, IS service catalogue, E-service catalogue, and public services catalogue.
- Common use E-services and IS services are technologically accessible from many access points – state and municipal portals, institution home pages, etc.
- Use of E-service envelope, when implementing asynchronous E-service.
- Safe mailboxes providing for office workers and inhabitants with possibility perform tasks, which are provided by E-service during implementation process.

ISIS solution conceptual description is shown on the Figure 1. Registers are related to integration software. For each data object, which is required for E-service realization, is necessary to work out XML schemas set.

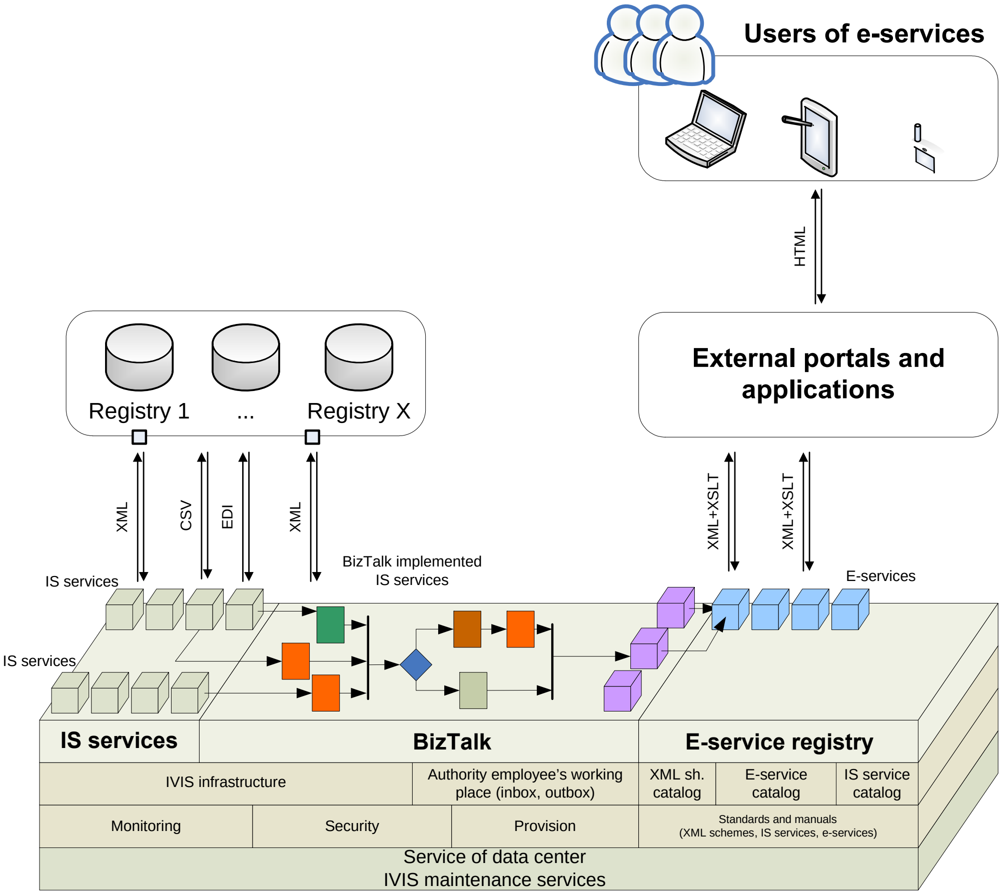


**Figure 1. Logical schema of ISIS solution basic elements**

This process is regulated by the corresponding standards and guidelines and is used for XML schema basic objects and types description. Data call from the corresponding register is performed by Web services help (in ISIS context these are named as IS services). During IS services calls occurs metadata sending, which describes the call. With the metadata register provider will receive the information, which is necessary for him to fill in his system's audit notes about a specific request. IS service development and using also is regulated by a corresponding standard. IS services are divided into two groups: Business IS services (includes some specific business activities, for example, get inhabitant's first name and last name by the personal code) and Integration IS services (includes some other Business IS service calls and E-service business logic). Implementation of Integration IS services usually is provided by the BizTalk server – one of the basic E-service infrastructure parts.

ISIS solution architecture comprises the following base elements:

- hierarchy of XML structures;
- IS service;
- e-service.

Platform for e-service implementation are made by several components which are:

- external portals and application which are made by:
  - e-service presentation layer;
  - external services and business logic libraries.
- ISIS infrastructure which is made by:
  - request service;
  - common use services of ISIS infrastructure;
  - integration IS services;
  - orchestration IS services (IS services made on BizTalk base what ensures operation of asynchronous and multi-step e-services);
  - business IS services.
- Providers of authentication and authorization information: management module of ISIS users and groups of users (MMUGU), inhabitants' accounts of state portal, safe electronic signature of JSC Latvia Post;
- ISIS applications;
- IS of state and municipalities.

Conceptual description of ISIS solution is available [8].

## 2.1 IS service catalogue

IS service catalogue is a part of ISIS architecture. It forms the unified repository for ISIS IS services which are used for E-service creation. In ISIS architecture with IS services is meant external interface by information resources. Information resource is register of every state agency or municipality, information system or database. The primary aim of each such register or IS is to ensure concrete, specific functions which the appropriate institution has to perform. IS and DB, which are used in institution are selected, developed and optimized specially for requirements of this specific institution, and they can be not oriented to possibility for others to receive information. But in the case if such possibility is intended it can be very specific and with different limitations. More detailed information about ISIS IS service is available [9].

## 2.2 E-service catalogue

E-service catalogue is a part of ISIS architecture. It forms the unified repository for ISIS E-services. E-service is on-line service ensured by state or municipality which is used by inhabitants, enterprises or state institutions. Applied architecture ensures possibility to provide the same e-service by use of unlimited number of entry points for example State portal, ISIS e-service portal and web pages of municipalities. More detailed information about ISIS E-service is available [**9**].

## 2.3 XML scheme catalogue

Data exchange between e-services and IS services is maintained by use of standardized and confirmed XML data structures which are included in centralized and publically accessible catalogue of XML structures. Organizational way of XML data structures is hierarchy, and data structure described in XML language can be included and re-used in other data structures as shown in figure 2. Centralized maintenance and well-considered development of hierarchy of XML data structures will ensure that each e-service or IS service will not need to have produced separate special data structure but more often used data objects (for example general personal data, general data of addresses etc.) will be repeatedly used in many of e-services and IS services. More detailed information about ISIS E-service is available [**9**].

## 2.4 Standard of metadata and e-service identification

By analogy with e-GIF, AGLS and other initiatives of e-management about metadata standard [7] set of Dublin Core (DC) elements were used [2]. Additionally XML scheme for verification of metadata elements was offered [6].

The second part of standard intends implementation of IETF RFC2141 [10] based identification system. Analysing existing options for resource identification it is formed scheme issuing unique identifiers according to URN principle what is regulated by IETF RFC4617 [1]. ISIS assigned identifier (according to URN syntaxes it is called as number) is written in the following form:

**URN:IVIS:100001:DOC-RCM-META**

Unique identifier of ISIS system has inherited all advantages of URN standard including:

- structural framework. Identifier has defined determinate structure (also in form of XML scheme) and is possible automated verification of it;
- readability. Identifier is understandable and usually includes understandable instructions for user on its source resource;
- uniqueness. Unique identifier is applied to information unit for unequivocal identification (XML schemes, documents and standards, ISIS IS services, e-services and e-service institutions);
- persistence. ISIS identifier stays unchanged from the time it is implemented by the time it is deleted. It is possible to delete ISIS code if it is useless, anybody does not use it or will not use.

Infrastructure of URN identifiers resolution has not yet established in the world scale. Global URN identifiers resolution system will be two-stage process. It will be formed as global identification register („identifiers space”) which will be able to find national, international or any other server which comprise information about registered URN and is responsible for information by definite NID (URN scheme). To such server will be re-addressed question to receive identifier resolution.

There are many functions used in URN ISIS system. With their assistance it is possible to receive information about XML scheme, IS service, e-service etc. One of ISIS URN application usage (ISIS URN resolution service) is described [9].

## 3 Logic programming paradigm

Logic programming is concentrated around basic mechanisms which are based on first-order predicate logic. It includes comparison with pattern, tree type data structure representation and automatic searching with rollbacks and cut offs. This limited complex of tools define powerful and flexible programming environment. That’s why logic programming is powerful tool for artificial intelligence and programming which is not based on numbers. This paradigm is based on first-order predicate logic and represents set of objects and dependencies between them. This means that for the conception of the representation of action use definition of mathematical relations or predicates. In logic programming attention is concentrated on declarative program meaning, on predicate definition not on procedure writing.

Logic programming fundamental concepts are:

- **Predicate** – truth function which is defined on concrete property or set of properties base. If object has this property (set of properties), then predicate equals to 1 (true), otherwise predicate value is 0 (false);
- **Fact** – true predicate, always true declaration;
- **Rule** – instructions, how to use to check concrete object functional precondition, to prognoze functionality in concrete situation and to make concrete conclusions.

Logic programming idea is needed to:

**1.** Describe set of declarations using formal language;

**2.** Use logical development system to get a solution.

### 3.1 Prolog language basics

The basic concepts of Prolog language are predicate, fact, rule and goal. In the logic of predicates the relationships between declarations and objects are overviewed. The predicate structure is shown on figure 2.

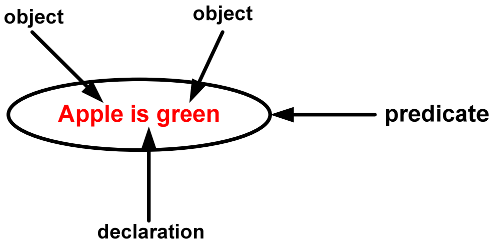


**Figure 2. Predicate structure**

In Prolog language the predicate on figure 2 can be defined as: is(apple, green). IS in this case is the functor of predicate. Apple and green are the objects (or components) of the predicate. The number of components in the predicate is not limited. Predicate with point at the end is fact. The components of the fact can only consist of letters, numbers and underlines. The first letter of the component name can be only lower case letter. Fact is always true predicate in Prolog language. Fact shows that some action is took place on object. When defining with the fact relations between the objects, it is necessary to think about the sequence of the components. The Prolog rule is used when it is necessary to show how one fact depends of another facts. Usually there is the left and the right parts in the rule. The right part of the rule is called head. It is the output part of the rule. The left part of the rule is called body. It is the conditional part of the rule. Parts of the rule are separated by the symbol ":-". Prolog language rule example is shown on figure 3.

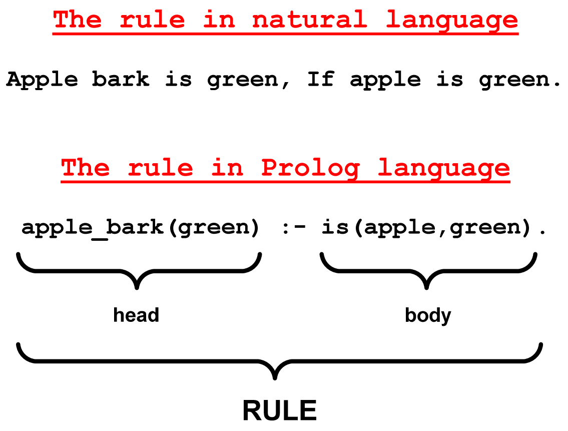

The rule in natural language
Apple bark is green, If apple is green.
The rule in Prolog language
apple_bark(green) :- is(apple,green).
head
body
RULE

**Figure 3. *Prolog* rule example**

Prolog goal – the formulation of the question for which the answer should be found [16]. Usually the question is the fact which need to check. If the fact or rule (are in the knowledge base) of the goal leeds to the goal then Prolog system gives answer „yes". That means that found fact is true, otherwise false. More information about Prolog language syntax is available [15].

The principle of the Prolog program execution – all is a tree. This principle from the very beginning is embeded in the Prolog environment. Programmers don't need to worry about the procedural part of the program, because of the declarative principle for the code writing in the program. This means that programmer need only to describe rules and facts and define goal. After that the program itself will care about goal consummation. The facts and the rules describe in the Prolog formal language which in many cases is very close to natural language. This opportunity is available because of the first layer predicate logic. More information about Prolog language execution principle of the programs is available [13].

Prolog language can be used:

- Development of the deductive data bases;
- Sphere of artificial intelligence:
    - o Development of the expert systems;
    - o Machine learning;
    - o And others
- Development of the relation data bases;
- Development of the games;
- Analyses of the graph.

### 3.2 Deductive data bases

Prolog language is very well known tool for solving problems related to deductive data base development. Deductive data base consist of two parts [14]:

- Part which contains facts (extencional part);
- Part which contains new rules which are based on new facts. These rules are based on extencional part and on user requests to data base (intencional part).

Deductive data base structure is shown on figure 4.

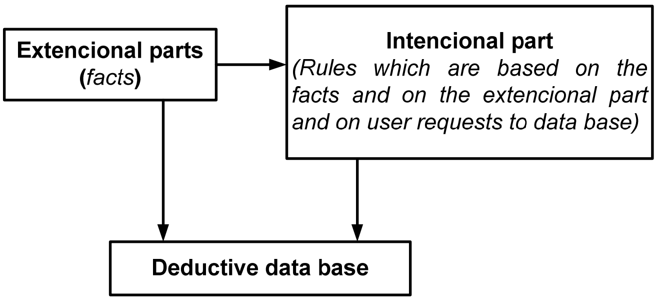

Extencional parts
(*facts*)
Intencional part
*(Rules which are based on the facts and on the extencional part and on user requests to data base)*
Deductive data base

**Figure 4. Deductive data base structure**

In deductive data bases recursion is very powerful tool. This tool is well implemented in Prolog. There is the direct relation between deductive data bases and knowledge data bases (are used in expert system development). Relation data bases are private case of deductive data bases. Using Prolog rules it is very easy to develop the extencional part of the data base. Using Prolog rules and goals it is very easy to develop the intencional part of the deductive data base.

## 4 Optimization for ISIS infrastructure

Each year there are new ideas of how to improve ISIS infrastructure. The special template for E-service development was created in 2008. The main advantage of this template that it is a compatible with ISIS infrastructure and was worked out using ISIS guidelines. A little bit later was developed ISIS URN resolution solution. Another one project is dedicated to E-service development optimization will be started as soon as possible. Last two solutions are described further in the paper.

### 4.1 ISIS URN Resolution

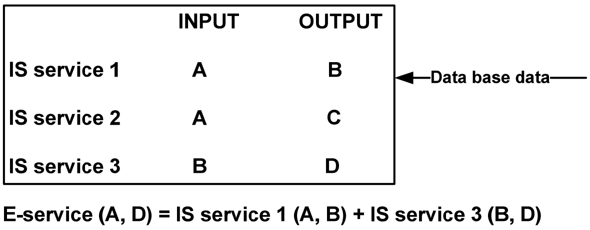
There are a lot of functions, which are executed using ISIS URN, in ISIS system. By ISIS URN is possible to get information about ISIS entity (XML schema, IS service, E-service etc.) to which this URN belongs. This information is metadata about specific ISIS entity. Metadata is based on DC elements set [2].

At present time only one authority issue such kind of URN to ISIS entities. In the future it is planned, that any authority will be able to issue ISIS URN according to agreement. This leads to the situation, where all important state entities will have globally unique identifier at state level – ISIS URN. All inhabitants will be able to get information about entities (resources), which provides specific authority. For this purpose it is planned to create ISIS URN resolution service. More information about this solution is available [8].

### 4.2 E-service development optimization

Different authorities (companies) using ISIS E-service development template create their own E-services. These E-services consist of IS services which are stored in IS Service catalogue. The number of such IS services growths day by day. Unfortunately some authorities create the same IS services (IS services with same input and output parameters) not using already existing IS services. The reason is that they can't find these IS services.

The main principle of the solution is shown on the Figure 5.



| | INPUT | OUTPUT |
|---|---|---|
| IS service 1 | A | B |
| IS service 2 | A | C |
| IS service 3 | B | D |

E-service (A, D) = IS service 1 (A, B) + IS service 3 (B, D)

**Figure 5. E-service development optimization**

Solution for such kind of situations is usage of deductive data bases and logic programming. This solution is the optimization for E-service development and works as follows:

1. Authority define input and output parameters for E-service.
2. Using logic programming software and deductive data base system searches for the solution.
3. If the result is positive and needed E-service can be created using existing IS services the list of these IS services is shown to the authority.

4. If the result is negative then authority can divide the E-service into subtasks and search solutions for them. This gives opportunity to develop only needed (missed) IS services and combine them with already existing.

There is a real E-service called EP00 in Latvia. It gives opportunity for inhabitants to find out the persons whose living place is declared in their property. It was the first ISIS infrastructure E-service in Latvia. It was developed, published and accessible in 2007. This E-Service works as follows:

1. Inhabitant selects EP00 on Latvian government E-service portal (www.latvija.lv) and signs in (after signing in inhabitant is authenticated and authorized for the portal; person's identifier is person code (for example, 010290-12345)).
2. The IS service *GetPersonEstateList* is called. Input parameter is person code. Output parameter is the list of cadastral values. These values identify person's, who called EP00, properties (for example, house or flat).
3. Person selects exact property for which he wants to get persons, who are declared for living in this place. The IS service *GetEstateObjUniqueAddressList* is called. Input parameter is cadastral value of the exact property. Output parameter information of the exact property (living place address inclusive).
4. The IS service *GetDeclaredPersonList* is called. Input parameter is living place address. Output parameter is list of the persons, who are declared (living) in this living place/property.

The graphical schema of the EP00 E-Service is shown on the Figure 6.

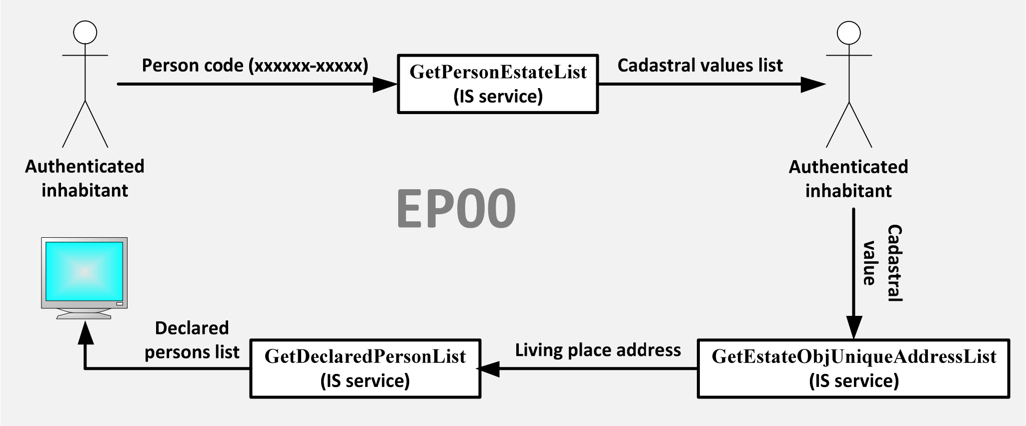


**Figure 6. E-Service EP00**

The described optimization (use of deductive approach for building E-services) gives opportunity quickly determine possible solutions (if they are) for building EP00. All is needed for that is:

1. Define input parameter person code;
2. Define output parameter list of persons;
3. Get list of possible solutions and manually determine the right one for EP00 creation. The solution will be the use/call of three IS services *GetPersonEstateList, GetEstateObjUniqueAddressList* and *GetDeclaredPersonList.*

## 5 Conclusions

The aim of E-service infrastructure creation is to implement an environment, in which it would be possible quickly and with minimal amount of programming work to implement E-services.

The created infrastructure of E-services provides:

- Increased quality and accessibility of state and municipal services (increased convenience and time economy for clients).
- Increased efficiency and examination of state and municipal institution activities (rational use of state and municipal funds).
- Ensured, advantageous and convenient service delivery procedure for clients.
- Ensured service accessibility, using various service delivery channels, which are appropriate for each client, such as ports, customer care centers, telephones, Internet, etc.

During realization of project some recommendations for usage and adaptation of SOA occurred:

- technological standard of integration has to be universal, it means that it has to be independent of different technologies of different software producers;
- integration standard has to be based on the best experience of the world and it has to comply with that what is now accepted and developed in international IT companies;
- integration software has to be scalable: with possibility to increase its performance without complicated re-programming works actually decreasing „down time" by the minimum because this software infrastructure will be a base for implementation of e-services and in the future are planned continuous increase either in electronization of services and intensity of their usage;
- integration software has to be with producers guaranteed development in perspective;
- it is necessary to use BPEL based solutions for integrated e-service development what ensure integration of several services and performance of asynchronous processes.

In more details positive and negative aspects of ISIS infrastructure will be researched during development of Doctorate work. In author's doctoral work it is planned to offer (work out) the new solutions which can be applied for ISIS infrastructure optimization at the State level.

## References

[1] A Uniform Resource Name (URN) Formal Namespace for the Latvian National Government Integration Project. 2006. Access via the Internet: http://www.ietf.org/rfc/rfc4617.txt

[2] Dublin Core Metadata Initiative. Access via the Internet: http://dublincore.org/

[3] E-pakalpojumu standarts v1.0. Access via the Internet: https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry41.aspx

[4] Internal portal "e-Gate of the Government" Access via the Internet: http://64.233.183.104/search?q=cache:S0cP9_ONKhwJ:www.ivpk.lt/en/report/report.pdf%20e-GATE&hl=en&ct=clnk&cd=2&gl=lt#27

[5] IS servisu izstrādes standarts v1.0. Access via the Internet: https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry10.aspx

[6] IVIS Metadatu standarta XML shēma. 14.03.2007. Access via the Internet: https://ivis.eps.gov.lv/EPAK/XSDK.WebSite/v1-7/Default.aspx?PagestatusID=XSDView&URN=URN:IVIS:100001:XSD-IVIS-Metadata-v1-0.

[7] Metadatu un e-pakalpojumu identifikācijas standarts v1.0. 12.01.2007. Access via the Internet: https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry38.aspx.

[8] **Semenchuk P., Kornienko Y., Narnicka V**. Databases and Information Systems. Tallinn, Estonia: Tallinn University of Technology Press, 2008, pp. 267-278.

[9] **Semenchuk P., Kornienko Y**. Service Oriented Architecture Implementation in Latvia Developed Software. *16th International Conference on Information and Software Technologies IT 2010*. Research Communications, Lithuania, Kaunas, 21.-23. April, 2010, pp. 91-98.

[10] URN syntax. 05.1997.Access via the Internet: http://tools.ietf.org/html/rfc2141.

[11] XML shēmu izstrādes vadlīnijas v1.0. Access via the Internet: https://ivis.eps.gov.lv/IVISPortal/files/folders/vadlinijas/entry8.aspx

[12] X-Road regulations – Internet. Access via the Internet: http://ftp.ria.ee/pub/x-tee/doc/X-Road_regulations.pdf

[13] Братко И. Алгоритмы искусственного интеллекта на языке PROLOG. – Москва·Санкт-Петербургб·Киев: Вильямс. 2004. 640 lpp.

[14] Дедуктивные базы данных. Access via the Internet: http://zeus.sai.msu.ru:7000/database/osbd/glava_119.shtml#_9_3_3

[15] Ин Ц., Соломон Д. Использование Турбо-Пролога. Москва: Мир. 1993. 608 lpp.

[16] Залогова Л. Язык программирования Пролог// Информатика. 2004. Nr.12. 7-11.lpp.