Formalization and Algebraic Verification of Legal Requirements

Alexander Godlevskyi[1], Aleksander Letichevskyi[1] , Vladimir Peschanenko[2], Oleksandr Letychevskyi[1], Maryna Morokhovets[1], Volodymyr Skobelev[1], Maksym Poltorackiy[2]

{a.godl49, aaletichevsky78, vpeschanenko, marina.morokhovets, mpoltorackiy}@gmail.com, lit@iss.org.ua, skobelevvg@mail.ru,

[1] Glushkov Institute of Cybernetics, National Academy of Science of Ukraine, Kiev, Ukraine
[2] Kherson State University, Ukraine

Abstract. This paper outlines the technology of the formalization and verification of legal text documents. A formal model of a legal document is presented as an ordered set of rules. Every rule is presented as a triple containing pre- and post-conditions, as well as process parts. All rules are ordered in Use Case Maps (UCM) diagrams. Such a formal model can be verified for inconsistency and incompleteness, which are the causes of gaps and ambiguities in legal requirements. Verification is implemented by means of algebraic computation and symbolic modelling. The formalization of legal requirements is performed with the use of an intermediate language. An example of the procedure of registration from the Tax Code of Ukraine is given for the illustration of the algebraic method implemented in the Insertion Modeling System (IMS).

Keywords: Legal Requirements, Algebraic Computation, Use Case Maps, Basic Protocols, Symbolic Modeling

1 Introduction

We examine the problem of the formalization and processing of legal requirements in the context of Ukrainian law, especially the Tax Code. The adopted laws and other artifacts of law activity often contravene the existing ones, and the causes of inconsistency could be deeply hidden in non-visible dependencies of rules and notions. These causes could result from such gaps as incompleteness or when precedent belongs to a situation that is not described in law. The other gap is ambiguities in notions or inconsistency when different consequences are applied to the same circumstances. The formalization of laws and the use of formal verification methods in the detection of such issues are among the ways to solve this problem. On the other hand, the existence of a formal model of laws allows an analysis of different precedents in terms of compliance with laws. Different interpretations of the precedent often lead to a wrong verdict, and formal proving could be a serious argumentation to cast away or to confirm such a situation.

To create a formal model, the existing formal languages used for formalization in different subject domains could be applied. Thus, a formal model of law could be created based on the technology of creating a formal model of requirements for software represented in natural language (NL). For the last 15 years, significant efforts have been directed toward developing models and methods intended to retrieve and analyze information from text sources [1]. However, the possibility of using this arsenal in legal document processing is essentially limited to the following circumstances.

It is well known that conflict detection plays an important role in the process of the analysis of legal texts. Existing approaches to this research problem for stable and sufficiently well-studied American and European legislation, in fact, are carried out under the following scheme [2].The source text, originally presented in an NL, is converted into some particular controlled natural language (CNL) [3]. The obtained text in the CNL with the use of some grammatical frame (GF) [14] is converted automatically into a text in some formal language CL [5,6]. The resulting text in the formal language CL is analyzed by various tools of CLAN [7], which can build counterexamples if conflicts are detected.

It should be noted that any used language CL must have the potential to represent such concepts used in legal texts as obligations, permissions, prohibitions, etc. It is also known that the inclusion of the time of the events in the analysis of information in the legal text documents is essential to the conclusions in the context of decision-making. Some attempts to implement this aspect were taken in [8-10] by marking time expressions with special labels.

Experiments on the analysis of contracts, agreements, and other small-scale legal documents showed that during processing, a number of significant difficulties, mainly in the implementation chain NL→CN→CL arise. In particular, substantial difficulties arise in the processing of synonyms for relevant legal concepts. A much more complex situation occurs in the analysis of a country's legislation.

It is well known that legal statements are interrelated and cannot be considered in isolation from each other. At the same time, the relationship between the statements in the legal text are presented by means of cross-references, which significantly affect the processes of structuring the organization, automatic generation of annotations for legal texts, and the extraction of the basic concepts (subjects, rights, duties, etc.).

Thus, the detection and resolution of cross-references in legal documents is one of the basic problems. Different approaches to the study of this problem for stable and sufficiently well-studied American and European legislation [11-15] show that algebraic methods, based on the development and analysis of some calculus [16-20], play a significant role in legal document processing.

2 Technology Overview

In the last decade the research in the field of legal document processing has also been conducted in Eastern Europe. It should be noted some attempts to present an XML model of a normative act (law) and to apply the XQuery language to perform applications in the processing of law were undertaken in Bulgaria [21]. Alongside the formal model proposed in [22], these attempts were intended to build a structured document (basic operations include inserting some new element, deleting, updating, and searching of the item). [23], demonstrating the possibility to apply the basic procedures used for the English-speaking legal texts to the analysis of the Polish legal acts.

US scientists have attempted to storage and processing of legal information in the format of GJXDM. The main problem was to create a set of specifications that would clearly define the structure of legal information data. The architecture of the regulatory legal act was represented in terms of xml, is that in the data dictionary used strictly defined vocabulary of names collected in an object-oriented data model (GJXDM).

One of the first expert systems was a system of "JUDITH" which was developed in 1975 in collaboration with experts from Heidelberg and Darmshtadsk universities. This expert system allows the lawyers to obtain expert conclusions in the field of civil law. Data of knowledge system consists of preconditions and executive files which indicate on the result that exist between sets of premises. Expert system "DSCAS" helps to analyze the law aspects of claims for additional costs compensation concerned with the difference in physical conditions of the place for building of the contract specified. Expert System provides the state employee with knowledge for lawsuit decision. Expert System DSCAS helps to analyze legal aspects of claims for additional costs related to the physical conditions award in case of construction specified in the contract[29]

Legal analysis system-helps the lawyers to conduct any legal analysis to deliberate crime cases. The legal knowledge theory, the facts of case are presented in the form of semantic networks. The system gives its user some findings, including its logic to the conclusion on which they are based. It substantiates the conclusion by links to judgments and supports legal documents. Quite powerful expert system is SARA, which helps lawyers to analyze decisions based on discretionary rules. The user tells the system some factors and their meanings. The system assigns weights of all the factors referring to regulation acts. Factors and solutions are presented in the form of frames. An iterative correlation method is used to calculate these weights [30-31].

One of the most powerful systems of such class is TAXAMAN-I and TAXAMAN-II systems. These systems specialize in tax law of England; the software tool architecture is implemented on the directory base. Designed intelligent system contains the output mechanism that is built on the model of logic programming. The output mechanism is built on the model of logic programming.

The system operation begins from an attempt to do the regulation of a high level by checking whether it was true premises. The system tests the truthfulness of such premises examining whether these backgrounds are true prerequisites. The process returns if it comes to a "deadlock". This process is called also as one that works on "inverse chain considerations" base, according to which the hypothesis trying to find the original data has to be proved.

Inference mechanism can work on the base of "direct chain of reasoning" in which arguments are conducted on the output data to the hypothesis. Subsystem of logical mechanism allows to process the information from its user and compare it with knowledge that is in the Base of knowledge. The user has to choose only one of the options, comparing the circumstances of the crime filed with components that lawmakers put in the law.

All expert systems are based on general law positions that in some way were structured and formalized, and also based on the personal perception of law, the legal system or legal norms, logic argumentation. In perspective, the expert legal systems can be used effectively in practice and systematization of legislation to solve the following important tasks:

1. Identifying of conflicting legal norms by expert interpretation.
2. To identify and eliminate legal gaps using similar rights or similar law;
3. Informal interpretation of unclear defined regulations, rules, concepts, principles.

The difference between EU "Byzantine law and acts" will certainly be to their own combination of expert opinions and estimates presented in audio. Combining different in its orientation methods and techniques of implementing information approach to learning objects mark the way to the discovery of new knowledge, previously inaccessible researcher in the empirical research in Byzantine and Slavic studies. According to the planned development of software support, which will support the storage and use of expert character data presented in both text and audio and video forms. In this regard, the main thing - the possibility of information-analytical finding and systematization of current scientific knowledge the information panel, which would meet modern standards of classification of scientific and technical staff of critical editions of historical sources [31].

Reference legal system "Garant" was developed in 1990. The system is provided with several exclusive tools. Among them:

The system contains the documents of the Russian legislation in the English version;

It helps to find the old version of the document and to refer to other documents of the time with the help of a click on the desired date in the comment;

Function "Related Documents" helps to place the hyperlinks to the documents of interest to you, as well as to examine documents that are similar to yours;

Chatting "online meeting" can correspond with colleagues and send links in real time;

Legislation in the schemes "section - a" skeleton "of the Russian legislation;

In the "Home Legal Encyclopedia you can find answers to common questions

"Guarantor" of the system is designed to work with the information base. This is the work with class information, search for documents, database updates. This box contains information of interest primarily economists and accountants. This block includes "tax calendar", informing about payment terms, the number of working days in the period information, and so on. Etc.

Search unit, this block contains a variety of search tools in the document information database; update block, in this block, the user can obtain statistical information about the information database and used to update a database, if necessary[33].

Bank "Expert application" includes special documents of the President, the Government, the Federal Assembly, in particular on: the state property and privatization, funding, government, legislative activities on international relations, staffing and other issues. Included narrow instruments of government, presented documents of Russian ministries and departments, including the Ministries of Finance, Central Bank, State Customs Committee, State Tax Service, the State Property Committee. The base system - documents relating to specific industries, specific areas and organizations, regulatory, technical and organizational regulations[30-31].

The first electronic filing cabinet for computer search of legal information was Credoc Belgian system, which made it impossible to read the full text of documents, but only with details (name, number, date of publication and so on). It consists of a bank of electronic documents [29].

One of the most famous full-text reference legal systems of the United States - LEXIS. Since 1980, the system became available to UK users, and from 1985 - Australia. It is responsible for more than 20 thousand-requests daily, basis of information gathered regulations, state and federal acts, including the full text of the US Constitution and all legal precedents USA [30].

Specifically focused on identifying and modeling relationships in a criminal group it allows you to record information that reflects their quality characteristics (corruption, family ties, etc.). Based on this analysis it cannot find strings that come from the immediate perpetrators to the organizers and sponsors of the crime [30-31].

A much more difficult situation occurs while processing legal documents in Ukraine, where legislation is only now being set and is subject to frequent changes. In contrast to the United States at this point, in Ukraine, there is no document like the Bluebook [24] or the US Association of Legal Writing Directors' (ALWD) Citation Manual [25], and the variety of cross-references (including the implicit ones) in Ukrainian law requires a thorough analysis.

Moreover, the implementation of the chain NL→CN→CL in addition to the abovementioned challenges, requires resolution of a variety of difficulties associated with the peculiarities in Ukrainian language grammar. This paper attempts to apply the technique of insertion modeling [26] and an algebraic approach, previously used for the verification of telecommunication models, to the analysis of legal texts, in this case to the Tax Code of Ukraine.

3 Formalization and Creation of Algebraic Model of Tax Code of Ukraine

Insertion modeling considers the interaction of agents and the environment. Agents are characterized by types, each of which having its own set of local attributes. The environment also contains global attributes. An interacting agent or agent inserted into the environment has a behavior that is described by the insertion function. This function is given by the rules of changing the states of agents and the environment. These rules describe local interactions and transitions that are called basic protocols [27]. Every basic protocol has a form $\forall x$ ($U(x)\rightarrow$<P> $V(x)$), where x is a list of parameters, U(x) – a precondition of basic protocol that defines the condition when the protocol can be applied, V(x) – a postcondition that defines transition of a system to a new state, and P – a process that illustrates the given transition. U(x) and V(x) are formulae of first-order logic constructing from predicates over integer, real, enumerated, and symbolic types, while P is a sequence of local actions and message passing. The state of a system is represented by means of simple attributes and attributes with parameters (functional attributes).

The basic concepts of insertion modeling are suitable to represent the notions and ideas used in the formulation of Tax Code articles. Laws are considered as requirements for the functioning of public institutions. These institutions constitute environment for taxpayers. There are executive and legislative bodies functioning within the environment that impose requirements on the operation of the agents: the various registries, supported by authorities, are considered as global attributes of the environment.

Authorities, firms, and businessmen (legal entities and persons) are acting as agents, interacting with each other and with the environment. However, the formalization of legal texts for further processing based on insertion modeling techniques has its own specifics. For example, the type of agent representing a taxpayer has the following attributes: name, set of activities, tax payment mode, status of payer, and others. Attribute values are changed by basic protocols that simulate the articles and clauses of laws regulating the legal actions of legal entities and authorities.

Document circulation and payments are simulated through a message passing mechanism between corresponding agents, where the types of messages are statements, requests, different templates.

Modeling of processes in time is carried out by two known mechanisms: timers and a global clock. Timers are used for the simulation of those regulations of the Tax Code, which are governed by restrictions similar to “answer must be given no later than 5 days after.” The global clock is to simulate restrictions associated with the implementation of actions, such as the timely payment of taxes.

Among the attributes of the agent, we could distinguish the agent status attributes of an enumerated type, specifying the variety of current agent modalities (obligations/restrictions/permissions).

For example, when an agent made another deal and increased its income, it may exceed a certain limit that obliges it to execute an additional registration.

Having applied for registration, it goes to an intermediate state, where the obligation is removed, but must wait for receipt of a positive response from the supervisory authority; then, the agent state will change to the “registered“ state that allows/obliges it to perform certain actions.

4 Example of Registration Procedure from Tax Code

As an illustration, below is considered a formal model of the procedure of the registration of a person as a payer of Value Added Tax (VAT), defined in articles 180-183 of the Tax Code of Ukraine. This model is a fragment of the general model that is being developed for the Tax Code.

The source for model development is the texts of the Tax Code articles in natural language. It anticipates the use of an intermediate language close to the natural. In this language, the legal items are reformulated in the form close to "if ... then ..." that is convenient for the creation of the next formal presentation as basic protocols.

Accordingly, in the development of the Tax Code model, one can identify the stages when a list of global attributes, types of agents, and their local attributes is formed. It is implemented with an exact reference to the Tax Code items or other laws referred therein. Of note is that the elements of environments and agents could be refined at later stages due to a better understanding of the essence of items. The next stage is the constructing of basic protocols that describe admissible action of agents.

Below, we present as example some items of article 183 in an intermediate language that will be used afterwards to create a formal model defining the explicit set of attributes. These items specify the actions of a person that should be registered as a taxpayer and the actions of local authorities.

We consider the interaction between agent Person with agent State Authority.

Rule 1. If Person (1.1) is not a single taxpayer (1.2) whose total income during last 12 months is more than 1,000,000 UAH, then Person (1.3) must be registered as a VAT payer.

Rule 2. . If Person (2.1) to be registered as a VAT payer (2.2) did not submit the application to State Authority before time T_CRITICAL, then Person (2.3) is responsible for a tax payment violation.

Rule 3. If Person (3.1) to be registered as VAT payer during (3.2) the period before T_CRITICAL is not completed, then Person (3.3) must submit the application to State Authority before T_CRITICAL.

Rule 4. If Person (4.1) must submit the application to State Authority, then Person (4.2) submits an application to State Authority.

Rule 5. If State Authority (5.1) has the application from Person, (5.2) there are no obstacles to refuse in registration; then, State Authority (5.3) registers Person as VAT payer.

5 Formal Model of Registration Procedure and its Verification

We consider Person and State Authority in a given procedure as types of interacting agents. Type of agent Person has attributes:

SingleTaxPayer – attribute of Boolean type that is equal to TRUE when it is a single tax payer;

MustRegisteredVAT – attribute of Boolean type that presents the status of Person when Person should be registered as VAT payer;

TotalIncome(t1,t2) – total income obtained by Person from time t1 to t2. It is a functional attribute, which maps (t1,t2) to an integer;

SubmitApplication – attribute of Boolean type that presents the status of Person when the application is submitted;

T_CRITICAL – time when the application shall be submitted. It is an integer-type attribute;

Responsibility – attribute of Boolean type that signalized responsibility of Person for wrong tax payment;

ShouldSubmitApplication – attribute of Boolean type that signalizes the necessity of application submitting;

NoObstacles – attribute of Boolean type that signalizes the absence of obstacles for registration as VAT payer.

The initial state of all agents could be presented as the below formula:

~SingleTaxPayer ∧ ~MustRegistredVAT ∧ ~SubmitApplication ∧ ~Responsibility ∧ ~ShouldSubmitApplication

We abstracted from other values of attributes considering it arbitrary. Time is considered as some integer attribute t changing during Person activity. The changing of time is fixed in the post-condition as the assignment statement t := t + a, where a is a parameter of the basic protocol. We consider days as the unit of time.

The order of events of the registration procedure could be presented as UCM notation, which is a subset of User Requirements Notation (URN) language, standardized by ITU (Z.151). The UCM diagram is an oriented graph containing constructs, such as its nodes with “responsibilities,” branches OR-fork and OR-join, references to other diagrams, and synchronization constructs. Basic protocols could also be presented without order and correspondingly without UCM. Below is the procedure of registration in UCM notation (Fig. 1).

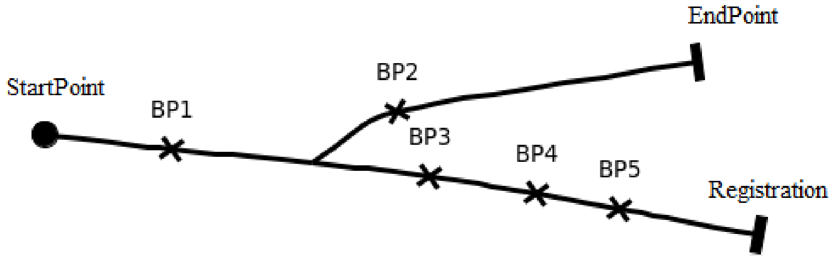

Fig. 1. Example of Registration Procedure presented as UCM-diagram

In the nodes of “responsibility” constructs, there are transitions associated with basic protocols. The list of basic protocols for agent P of type Person and agent A of type Authority is given below:

BP1 (1.1, 1.2, 1.3). $\forall a$, $(a > 0) \wedge \sim$SingleTaxPayer $\wedge$ (TotalIncome(t - 365,t) > 1000000) $\wedge$ $(t > 365) \rightarrow$ <Status changed: Must be Registered as Payer of VAT> t:=t + a; MustRegisteredVAT

BP2 (2.1, 2.2, 2.3). $\forall a$, $(a > 0) \wedge$ MustRegisteredVAT $\wedge \sim$SubmitApplication $\wedge$ (t > T_CRITICAL) $\rightarrow$ <Status Changed:Responsibility for Incorrect Tax Payment> Responsibility

BP3 (3.1, 3.2, 3.3). $\forall a$, $(a > 0) \wedge$ MustRegisteredVAT $\wedge$ (t <= T_CRITICAL) $\rightarrow$ <Status Added: Must Submit Application> t:=t + a; ShouldSubmitApplication

BP4 (4.1, 4.2). $\forall a$, $(a > 0) \wedge$ ShouldSubmitApplication $\rightarrow$ <Agent P: Submit Application to Agent A> t:=t + a; SubmitApplication $\wedge \sim$ ShouldSubmitApplication

BP5 (5.1, 5.2, 5.3). $\forall a$, $(a > 0) \wedge$ NoRestriction $\wedge$ SubmitApplication $\rightarrow$ <Agent A: Register Agent P> VATTaxPayer $\wedge \sim$MustRegisteredVAT

Every basic protocol presents the rule with a corresponding number expressed in natural language in a previous section. Traceability to rules pre- and post-condition is defined in brackets. This system of basic protocols could be symbolically modeled from the initial formula by means of the application of a predicate transformer function [28].

$$E_{new} = Pt(V\wedge Eold, U)$$

The predicate transformer Pt computes the new state of the environment E_{new} and agents from the old environment state Eold correspondingly to pre- and post-conditions V and U. The system of basic protocols is inconsistent if it contains non-determinism or rules where under the impact of the same conditions, two different changes to the environment and agents are possible.

Regardless, we should consider different non-determinisms, such as undesirable non-determinism, which belongs in our case to “necessary” actions, such as to pay corresponding taxes. The other kind of non-determinism belongs to admissible non-determinism, which presents the “desirable” actions of agents, for example, to select the mode of tax payment. Consistency of two basic protocols can be presented by the below formula:

(1) $\forall$ x, y(U(x) $\wedge$ U(y)) = 0,where x and y – are basic protocols and Ux, Uy corresponding preconditions.

The system is inconsistent if the below formula

$$(2) \sim (Pt\ (U(x),V(x)) \leftrightarrow Pt(U(y),V(y))) \wedge (U(x) \wedge U(y))$$

is satisfiable. U(x), U(y), V(x), and V(y) are corresponding pre- and post-conditions, while Pt designates the predicate transformer. When basic protocols are presented in UCM notation, formulae (1) and (2) are defined for basic protocols that are simultaneously applicable correspondingly to UCM diagrams.

The given system is consistent. Let us consider the example when a new law was submitted where some group of manufacturers that are not single tax payers should be registered as single tax payers if they have a total income more than 1,000,000 UAH. It is formalized by the following basic protocol.

BP6. $\forall a$, $(a > 0) \wedge$ SpecialGroup $\wedge \sim$SingleTaxPayer $\wedge$ (TotalIncome(t - 365,t) > 2000000) $\wedge$ $(t > 365) \rightarrow$ <Status changed: Must be Registered as Single Tax Payer > t:=t + a; MustRegisteredSingleTax

There are two Boolean variables: SpecialGroup, defined belonging to the group of manufacturers, and MustRegisteredSingleTax, a defining status of necessity to register as a single tax payer. Here is ambiguity, and formula (2) is satisfied for protocols BP1 and BP6. It reflects the case when a member of the special group of manufacturers earned a total income more than 1,000,000 UAH. If he or she obtained more than 1,000,000 UAH, two rules are applicable and it is unknown whether he or she should be registered as VAT payer or as single taxpayer. These issues could be corrected if rule BP1 is modified by adding ~SpecialGroup to the precondition.

This system is incomplete. We can see the reachable incompleteness in basic protocol BP5. What can be done if NoRestriction is equal to FALSE? In this case, the state VATTaxPayer is unreachable and Person will never be registered as a VAT payer Completeness is presented by formula:

$$(3)\ \ \vee U(x) = 1$$

where U(x) represents preconditions of all basic protocols. When basic protocols are presented in UCM notation, formula (3) is defined for basic protocols that are simultaneously applicable correspondingly to UCM diagrams.

Incompleteness could be a source of the wrong treatment of the law. For example, if there is a precedent that a person with an income more than 1,000,000 UAH has a registered application, but there are some causes to refuse in registration from the side of the Authority. The payer continues activity but does not pay the VAT. Let the Tax Inspectorate charge him with a wrong tax payment. Formally, it means that Responsibility = TRUE. On the other side, he submitted the application for registration that is SubmitApplication = TRUE.

Providing symbolic modeling from the initial state of agents and the environment, we can prove the unreachability of state Responsibility = TRUE $\wedge$ SubmitApplication = TRUE. The given formal proof could be used to justify the negation of the responsibility of agent Person.

6 Conclusion

The authors try to consider to a great degree the detailing of the formalization of the Tax Code in terms of VAT with a view to check it for inconsistency and incompleteness. There is also a goal to consider precedents that caused conflicts among VAT payers and Tax Inspection and give a formal foundation of the rightness or mistakenness of subjects.

The formalization is being performed in the Insertion Modeling System (IMS) with the use of a deductive solving system. It is anticipated that the shortcomings of the Tax Code will be detected and a corresponding technology of the formal verification of law will be presented in modern legal studies.

References

1. C.D. Manning, P. Raghavan, H. Schutze, “Introduction to Information Retrieval,” Cambridge University Press, 2009, 544 p.

2. K. Angelov, J. Camilleri, G. Schneider, ”A framework for conflict analysis of normative texts written in controlled natural language,” The Journal of Logic and Algebraic Programming, 2013, vol. 82, №5, pp. 216-240.
3. A. Wyner, K. Angelov, G. Barzdins, D. Damljanovic, B. Davis, N. Fuchs, S. Hoefler, K. Jones, K. Kaljurand, T. Kuhn, M. Luts, J. Pool, M. Rosner, R. Schwitter, J. Sowa, “On controlled natural languages: properties and prospects,” LNCS/LNAI, 2010, vol. 5972, pp. 281–289.
4. A. Ranta, “Grammatical Framework: Programming with Multilingual Grammars,” CSLI Publications, Stanford, 2011.
5. C. Prisacariu, G. Schneider, “A Formal Language for Electronic Contracts,” LNCS, 2007, vol. 4468, pp. 174–189.
6. C. Prisacariu, G. Schneider, “CL: An Action-based Logic for Reasoning about Contracts,” LNCS, 2009, vol. 5514, pp. 335–349.
7. http://www.cs.um.edu.mt/~svrg/Tools/CLTool
8. F. Schilder. “Event Extraction and Temporal Reasoning in Legal Documents” LNAI, 2007, vol. 4795, pp. 59–71.
9. V. Naik, V. Guda, I. Srujana. “Reasoning in legal text documents with extracted event information,” International Journal of Computer Applications, 2011, vol. 28, №7, pp. 8-13.
10. K. Ramakrishna, V. Guda, B.P. Rani, V. Chakati, “A Novel model for timed event extraction and temporal reasoning in legal text documents,” International Journal of Computer Science & Engineering Survey, 2011, vol. 2, №1, pp. 39-48.
11. M. Palmirani, R. Brighi, M. Massini, “Automated extraction of normative references in legal texts,“ Proc. of the 9th International Conference on AI and Law, 2003, pp. 105–106.
12. E. de Maat, R. Winkels, T. van Engers. “Automated detection of reference structures in law,“ Proc. of the 19th Annual Conference on Legal Knowledge and Information Systems, 2006, pp. 41–50.
13. N. Kiyavitskaya, N. Zeni, T. Breaux, A. Ant´on, J. Cordy, L. Mich, J. Mylopoulos, “Automating the extraction of rights and obligations for regulatory compliance,“ Proc. of the IEEE 16nd International Requirements Engineering Conference, 2008, pp. 154-168.
14. M. Hamdaqa, A. Hamou-Lhadj, “An approach based on citation analysis to support effective handling of regulatory compliance,“ Future Generation Computer Systems, 2011, vol. 27, №4, pp. 395–410.
15. O. Tran, M. Le Nguyen, A. Shimazu, “Reference resolution in legal texts,“ Proc. of the 14th International Conference on AI and Law, 2013, pp. 101–110.
16. M. Adedjouma, M. Sabetzadeh, L.C. Briand. “Automated detection and resolution of legal cross references: approach and a study of Luxembourg’s legislation,“ Proc. of the IEEE 22nd International Requirements Engineering Conference, 2014, pp. 63-72.
17. H. Cunningham et al. “Developing language processing components with GATE Version 7 (a User Guide),“ http://gate.ac.uk
18. D. Beyer, A. Noack, C. Lewerentz, “Efficient relational calculation for software analysis,“ IEEE Transactions on Software Engineering, 2005, vol. 31, № 2, pp. 137–149.
19. P. Rosso, S. Correa, D. Buscaldi, “Passage retrieval in legal texts“ Journal of Logic and Algebraic Programming, 2011, vol. 80, №3-5, pp. 139-153.
20. P. Quaresma, T. Goncalves, “Using linguistic information and machine learning techniques to identify entities from juridical documents“ LNCS, 2010, vol. 6036, pp. 44-59.
21. V. Peychev, “XML model for legal documents“ Problems of Engineering Cybernetics and Robotics, 2004, vol. 54, pp. 86-91.
22. V. Peychev, “Legal document – a formal model“ Problems of Engineering Cybernetics and Robotics, 2005, vol. 55, pp. 64-70.

23. W. Cyrul, T. Pełech-Pilichowski, P. Potiopa, “Problems of automatic processing and analysis of information from Polish legal texts,“ Proc. of the IEEE Federated Conference on Computer Science and Information Systems, Poland: Wrozlaw, 2012, pp. 939–943.
24. The Bluebook: A Uniform System of Citation, 19th Ed., Harvard Law Review, 2010.
25. Association of Legal Writing Directors and D. Dickerson, The ALWD Citation Manual, 4th Ed., Aspen Publishers, 2010.
26. A. Letichevsky and D. Gilbert, “Interaction of agents and environments,“ in: *Resent trends in Algebraic Development technique, LNCS 1827* (D. Bert and C. Choppy, eds.), Springer-Verlag, 1999.
27. A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V.Kotlyarov, T. Weigert, “Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications,“ Computer Networks, 47, 2005, pp. 662-675.
28. A. Letichevsky, O. Letychevskyi, V. Peschanenko, T. Weigert, “New predicate transformer for symbolic modeling”, Proc. of Workshop PAS-2014 in CAV conference.