

USE OF METHODS OF ALGEBRAIC PROGRAMMING FOR THE FORMAL VERIFICATION OF LEGAL ACTS

Volodymyr Peschanenko, Maksim Poltorackiy

This article briefly describes the programmable tool for the analysis of a normative legal document. A mechanism for checking legal requirements is presented. The model of the legal document is proposed in the form of a set of special rules. Verification is provided by means of algebraic programming and methods of symbolic transformation. This approach allows us to analyze the legislative base of structural and logical errors, check the contradictions, completeness and integrity of legal acts. Presently, the mechanism of cluster analysis of text, which makes it possible to identify the frequency of occurrence of various vague language constructs.

Key words: law, algebraic programming, legal requirements, algebra of behavior.

У цій статті коротко описується програмний інструмент для аналізу нормативно-правового документа. Існує механізм перевірки законних вимог. Модель правового документа представлена у вигляді набору спеціальних правил. Верифікація забезпечується за допомогою алгебраїчного програмування та методів символічного програмування. Цей підхід дозволяє проаналізувати законодавчу базу на наявність структурно логічних помилок, перевіряти правові вимоги на протиріччя, повноту та цілісність.

Ключові слова: закон, алгебраїчне програмування, юридичні вимоги, алгебра поведінки.

В этой статье кратко описывается программный инструмент для анализа нормативного правового документа. Представлен механизм проверки юридических требований. Модель юридического документа представлена в виде набора специальных правил. Верификация обеспечивается с помощью алгебраического программирования и методов символического преобразования. Такой подход позволяет нам анализировать законодательную базу на наличие структурно-логических ошибок, проверять противоречия, полноту и целостность правовых актов.

Ключевые слова: закон, алгебраическое программирование, юридические требования, алгебра поведения.

Introduction

The regulatory framework is prone to frequent changes, the recently enacted laws may contradict existing laws, which may lead to possible wrong verdicts.

The situation is also possible when two laws with different consequences can be used for a single legal case. An example of such a legal case is a precedent: "Convicted or a precedent in favor of a taxpayer". A large number of cross-references between normative legal acts makes the analysis of the legislative base of Ukraine more complicated.

The formalization of laws and the use of methods for formal verification of the model of legal acts is one of the ways to solve this problem. To create a formal model, formal languages are used that can be used to formalize information in different areas. However, the possibility of using this arsenal in the processing of legal documents is significantly reduced. Any official language used to formalize a legal text should operate with legal terms (obligations, permits, prohibitions). The complexity of this method – the process of formalization – is a rather lengthy process and requires the interaction of two specialists: a specialist in law and a knowledge engineer (specialist in information technology).

The issue of the possibility of using expert systems in legal activities was repeatedly noted by many authors, discussions on this issue have been conducted since the 60s of the twentieth century. Most expert legal systems are built on general legal provisions that must be structured and represented in such a system; the functioning of systems related to the following problems:

- problems of transferring knowledge from human experts to a computer system;
- the problems of submission of knowledge, that is, reconstruction of an array of knowledge in a particular area of law and its presentation as a structure of knowledge;
- problems of the use of legal knowledge.

Next, consider expert systems and methods of presenting legal information.

Technology Overview

1. CASE-LAW MODEL

The Shyster system is based on case law. Uses a case-law model for presenting legal texts.

The functionality of this expert system can be divided into two parts:

1. Legal search system

2. System of legal analysis. An important feature of this system is the ability to provide certain legal conclusions and arguments for the lawyer.

The laws are presented in the form of a specialized specification language. The main idea of this system is based on a case-law description of legal cases, that is, there is a knowledge base in the system.

The system considers existing cases and on the basis of old judgments will try to predict the outcome of a new case. An important feature of this system is the ability to provide certain legal conclusions and arguments for the lawyer. This system is positioned as a decision support system for lawyers. There is no functional possibility of verification of legal norms.

2. PLANNING AND CALCULATION OF STRATEGIES

Expert system TAXADVISER [1], used in the taxation process. This system is based on certain mathematical functions and strategies that the lawyer has already formulated on the basis of legal materials.

This approach is based on the planning and calculation of strategies that were obtained by legal experts. The peculiarity of the approach in this system is the use of the strategy that the lawyer expert provided.

3. LOGICAL PROGRAMMING

In these expert systems, a model is used that is built on the basis of a certain number of rules of the type: IF-THEN.

A set of rules is called a "search space", it is compared with a set of facts. In logical programming it is supposed to search for the correspondence of a condition to facts and on the basis of "direct" or "reverse" output reach a certain logical conclusion. This technology is also used by the following systems: Schlobohm, LEX.

4. XML AND XQUERY LANGUAGE

Also, interesting attempts were made to develop a model of a normative legal document in the form of an xml document, which in turn would allow using the XQuery language to process this type of document [2, 3]. American scientists tried to present regulatory legal acts in the GJXDM format, the main problem of this approach is the creation of a certain specification that clearly defined the structure of this legal information. The architecture of the normative legal act was presented in terms of an xml document with a strictly distinguished dictionary of names collected in the GJXDM object-oriented information model.

Such an approach did not give an opportunity to analyze normative legal acts, but only allowed to implement the mechanism of requests for processing a document of this type.

5. DEONTIC AND FUZZY LOGIC

Interesting approaches for presenting the legal text and further use in jurisprudence for legal reasoning was the decision to use deontic and indistinct logic.

The representative of the deontic approach was the speech LLD, which was based on the principles of deontic logic and had certain advantages for legal application in the field of legal argumentation. LLD, tried to present legal laws in graphic terms and used indistinct categories that are inherent in this logic [4].

(Own O1 (Actor A) (Property P)),
(Own 'Own-2
(Actor 'John) (Property 'Book-3)),
(Own - (Actor 'John)
(Property 'Book-3))

Atomic Formula in LLD language [4]

The approach using indistinct logic is another attempt to move from very approximate terms that are used in legal texts to clear terms that are inherent in logic and can be further computer processed.

6. ONTOLOGY IN LAW

One should also mention the concept of building a model of a legal document using the concept of anthologies.

The basic concept of anthologies can be used to represent normative legal acts in the form of owl-files, this structure will allow to implement more complex textual queries and use the concept of SemanticQuery.

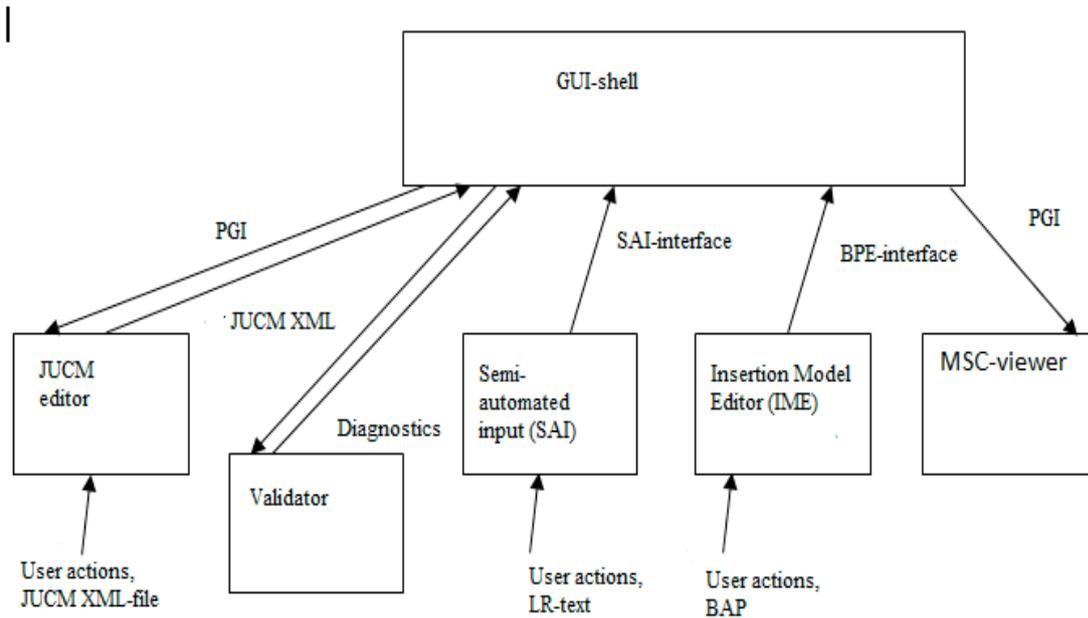
Each of the approaches to constructing a formal model of a normative legal act is quite interesting and has certain advantages, but none makes it possible to verify the formal properties of the model of legal requirements and analyzed the legislative basis for logical and structural errors.

The program that we are developing will provide an opportunity to analyze the legal framework and solve the following problems:

- Checking the legal framework for structural and logical correctness;

- Checking the rule of law for the presence of contradictions in the formation of the regulatory framework.
- Check the model of laws for the completeness of the description and the integrity of the model.

This program will not only provide an opportunity to check the existing legal and regulatory framework, but also analyze the new laws that will be added. The application architecture can be represented as follows:



Figur. HLD-scheme of GUI

Let's briefly consider the main components of the program:

1. jUCM editor receives jucm-file with an .xml extension, this file is for creating a UCM diagram.
2. The validator processes the BAP files, and then returns the BAP file diagnostics.
3. MSC Viewer – graphical display of MSC chart elements. Graphically displays instances, conditions, alternatives, altas, akshyns, messages between instances.
4. Insertion Model Editor (IME) – A custom editor for creating basic protocols.
5. Statistic – a module that will provide an opportunity to look at the descriptive characteristics of laws, see what structural components affect the outcome of the applicability or inapplicability of this law.

At the heart of the approach for calculating the effect, we will use ogit., Probit. model and displayed on a special structural diagram. Also, in the statistics module there is the possibility of isolating the body of knowledge of the source text, as well as the possibility of constructing a cluster analysis of the text, building a dendrogram of the original legal text.

Using a combination of such approaches will make it possible not only to verify structurally logical errors, but also to evaluate the text of the regulatory framework from the point of view of static text analysis. Such an approach will make it possible to more effectively use quantitative methods, especially attention is paid to identifying the frequency and distribution of individual language structures or their groups, as well as stochastic dependencies between them or between groups of such designs.

To identify structural and logical errors, as well as to check the regulatory framework for formal properties: inconsistency, completeness and integrity of the model, methods and technologies of insertion modeling (IM) are used. Next, consider the concept of using IM technology as the main approach for presenting the legal framework, as well as submitting its verification. The main feature of our approach is the use of methods of symbolic transformations and algebraic programming in the IM system. Ininsertion modeling was developed at the V.M. Glushkov Institute of Cybernetics [5].

The basic concept of insertion modeling can be used to interpret the legislative base. Laws can be considered as standard requirements for functioning. Firms, entrepreneurs and governance can act as agents, various legislative bodies

impose certain requirements on the functioning of agents in the environment. The type of agent that represents the taxpayer can have the following attributes: name, tax payer status, tax payment regime. In the case of fraud and various processes over time, a timer and a global clock will be used. The global clock will be used for restrictions related to the implementation of actions, and the timers will be used to process requests for legal requirements. The behavior of the system can be described as a system of basic protocols. The basic protocol is the expression

$$\square x (U(x) \rightarrow \langle P \rangle V(x)),$$

where $U(x)$ – is the premise that determines the state when the protocol can be applied, $V(x)$ – post condition, which determines the transition of the system to a new state, P – is the process that illustrates this transition. The basic idea of the theory of basic protocols is set out in a system that is specified, represented as a composition of agents and media [6].

The environment is an agent that has a immersion function. The environment is a set of $\langle E, C, A, Inc \rangle$, where E – is the set of environmental states, C – is the set of actions of the environment, A – is the set of actions of agents that are immersed in the medium, $F(A)$ – is the complete behavior algebra of the agent with the set of behaviors and $Ins : E \times F(A) \rightarrow E$ – immersion function. Thus, some "E" environment admits an immersion of any agent with the set of actions "A" [7].

The taxpayer can be represented in the system of insertion modeling in the following form:

```
agent TaxPayer : obj (
  Activity : obj (
    Code : int,
    VATpayer : bool,
    .....
  )
  Register:obj(
    Voluntary:bool,
    Must:bool,
    Perfomed:bool,
    TotalIncome : int,
  )
  .....
)
```

Using the basic principles of insertion modeling, it is possible to present legal requirements in the form of basic protocols, and axioms of insertion modeling. Let's consider a fragment of the normative legal act that defines the requirements for the VAT payer of Ukraine. For convenience of formalization of the normative legal act, we will present the text in the form of an if-then rule. Then the rule will look like this:

Rule: If "A person is not a registered taxpayer and for the past 12 months, a person has received an income of more than 1,000,000, then the person must register as a VAT taxpayer".

This statement can be written in the IMS in the following way (since in the text of the law the fact of a certain action is traced, formalization is represented as a basic protocol):

```
forall(x:TaxPayer)(!(TaxPayer.VATPayer)&&(TaxPayer.TotalIncome(t 365,t) > 1000000)&&(t > 365)
-< Must registration TaxPayerVAT > (TaxPayer.VATPayer)).
```

Next we will consider a fragment of the text, Law180.

Having considered the interpretation of the law, it is not difficult to understand that the requirement for a taxpayer can be interpreted by a sequence of logical operations (conjunctions, disjunctions and equivalence), since at the clause of law 180 the fact of action is not formally indicated, then the formalization is performed as inertial modeling axioms. We give an example of the axiom from the law 180:

- A180_1_1 <=>

(((x.commercialActivity.status = GONNA_BE) || (x.commercialActivity.status = DOES)) &

(x.registrationVAT.status = DOES) &x.registrationVAT.decisionVoluntaryVAT))
- art_180_1_2 <=>

(x.registrationVAT.necessity || (x.registrationVAT.status = DID))

....

- $A180_1_5 : (A180_1_5 \Leftrightarrow (x.otherActivity = CONTROL_ESTATE))$

Then completely a fragment of the law, can be interpreted as follows (as a certain logical sequence of axioms):

- $art_180_1 \Leftrightarrow (x.TaxPayer \Leftrightarrow (art_180_1_1 \vee art_180_1_2 \vee \dots \vee art_180_1_5))$.

This approach can be applied to the entire tax code of Ukraine, but in the process of formalization we found the following laws, the text of which is quite difficult to present in a formalized form, because their design can be interpreted in different ways. In the following we consider the following laws:

186.1. The place of supply of goods shall be:

186.2.1. the place of actual supply of services related to movable property, namely:

- ancillary services in transportation activity: loading, unloading, reloading, warehouse processing of goods and other similar services;
- actual location of services in culture, art, education, science, sport, entertainment or other similar services, including without limitations services by organizers of activities in the said areas and services provided for organization of paid exhibitions, conferences, scientific workshops and other similar events [8].

The construction of these laws is a counter-version, formulations like: or other, etc. are difficult to interpret with formal languages.

In order to formalize such a law, it is necessary to clearly understand what are similar services within the framework of Article 180, paragraph 180.2.1. The time given for the development of adequate formalization within these laws should clearly interpret the meaning of this expression, because in this form the meaning can be subjective.

Conclusions

In this article, briefly describes the functionality of the program based on methods and techniques of insertion modeling. The technology of presenting legal information using IMS technology is described.

An important feature of this approach is the use of this technology for formal verification of the legal framework. This approach makes it possible to use the IMS methodology to check the regulatory framework on: inconsistency, completeness and curative.

The use of methods of cluster analysis of text provides a unique opportunity to assess the initial text of the regulatory framework in terms of static text analysis.

The possibility of isolating the knowledge base of the normative legal base document and the methods of cluster analysis will make it possible to use quantile methods more effectively to determine the frequency of individual language constructions or groups of such constructions.

Proceeding from the analyzed literature, I want to emphasize that not one of the technologies makes it possible to analyze the text of a normative legal document for such a wide class of problems. Already at the initial formalization process, you can analyze and identify weaknesses in the regulatory base of Ukraine. The implementation of such a tool is solved by the following class of tasks:

- Review of regulatory legal documents on the possibility of revealing formal and logical errors.
- Construction and verification of the normative legal document model for contradictions, completeness and integrity.
- Analysis of the text of the source document to determine the frequency of the appearance of ambiguous language text constructs, as well as highlight the body of knowledge of the source text.

This program module will give an opportunity to elicit controversial constructions of the legislative base of Ukraine, to identify possible structural and logical errors, as well as possible contradictory constructions in the legislative framework.

References

1. Michaelsen R.H. 1984. An expert system for federal tax planning. *Expert Systems*, 1(2). P. 149–167.
2. Peychev V. 2004. XML model for legal documents. *Problems of Engineering Cybernetics and Robotics*. 54. P. 86–91.
3. Peychev V. 2005. Legal document – a formal model. *Problems of Engineering Cybernetics and Robotics*. 55. P. 64–70.
4. McCarty L.T. 1989, May. A language for legal Discourse I. basic features. In *Proceedings of the 2nd international conference on Artificial intelligence and law* (P. 180–189). ACM.
5. Letichevsky A.A., Letychevskiy O.A. and Peschanenko V.S. 2011, June. Insertion modeling system. In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics* (P. 262–273). Springer, Berlin, Heidelberg.

-
6. Letichevsky A., Kapitonova J., Letichevsky Jr.A., Volkov V., Baranov S. and Weigert T. 2005. Basic protocols, message sequence charts, and the verification of requirements specifications. *Computer Networks*, 49(5). P. 661–675.
 7. Letichevskii A.A., Kapitonova Y.V., Volkov V.A. and Vyshemirskii V.V. 2003. Insertion programming. *Cybernetics and Systems Analysis*, 39(1). P. 16–26.
 8. Ukrainy P.K. Tax Code of Ukraine] vid 02.12. 2017, (zi zminamy i dopovnenniamy). zakon4. rada. gov. ua.
 9. Popple J. 1996. A pragmatic legal expert system. Dartmouth (Ashgate).
 10. Michaelsen R.H. 1984. An expert system for federal tax planning. *Expert Systems*, 1(2). P. 149–167.

Література

1. Michaelsen R.H. 1984. An expert system for federal tax planning. *Expert Systems*, 1(2). P. 149–167.
2. Peychev V. 2004. XML model for legal documents. *Problems of Engineering Cybernetics and Robotics*. 54. P. 86–91.
3. Peychev V. 2005. Legal document – a formal model. *Problems of Engineering Cybernetics and Robotics*. 55. P. 64–70.
4. McCarty L.T. 1989, May. A language for legal Discourse I. basic features. In *Proceedings of the 2nd international conference on Artificial intelligence and law* (P. 180–189). ACM.
5. Letichevsky A.A., Letychevskiy O.A. and Peschanenko V.S. 2011, June. Insertion modeling system. In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics* (P. 262–273). Springer, Berlin, Heidelberg.
6. Letichevsky A., Kapitonova J., Letichevsky Jr.A., Volkov V., Baranov S. and Weigert T. 2005. Basic protocols, message sequence charts, and the verification of requirements specifications. *Computer Networks*, 49(5). P. 661–675.
7. Letichevskii A.A., Kapitonova Y.V., Volkov V.A. and Vyshemirskii V.V. 2003. Insertion programming. *Cybernetics and Systems Analysis*, 39(1). P. 16–26.
8. Ukrainy P.K. Tax Code of Ukraine] vid 02.12. 2017, (zi zminamy i dopovnenniamy). zakon4. rada. gov. ua.
9. Popple J. 1996. A pragmatic legal expert system. Dartmouth (Ashgate).
10. Michaelsen R.H. 1984. An expert system for federal tax planning. *Expert Systems*, 1(2). P. 149–167.

About the authors:

Peschanenko V.S.

Doctor of physical & mathematical sciences, assistant professor
Professor of the department of Informatics, Software Engineering and
Economic Cybernetics of the Kherson State University.

Articles: 23 (Scopus), total article: 53.

h-index: 3 (Scopus).

<http://orcid.org/0000-0003-1013-9877>.

Maksim Poltoratskiy (to correspond)

PhD student of the Department of Informatics,
Software Engineering and Economic Cybernetics of Kherson State University.

Articles: 6 (Scopus), total article: 11.

h-index: 2 (Scopus).

<http://orcid.org/0000-0001-9861-4438>.

Affiliation:

Kherson State University,
Str. Universytets'ka, 27, 73000, Kherson, Ukraine.
Tel: +38 (095) 324 1557.
E-mail: vladim@ksu.ks.ua.

Tel: +38 (050) 668 6238.

E-mail: mpoltorackiy@gmail.com