Krassimir Markov, Vladimir Ryazanov,
Vitalii Velychko, Levon Aslanyan
(editors)

New Trends in Classification and Data Mining

I T H E A
SOFIA
2010

Krassimir Markov, Vladimir Ryazanov, Vitalii Velychko, Levon Aslanyan (ed.)

New Trends in Classification and Data Mining

ITHEA®

Sofia, Bulgaria, 2010

First edition

Recommended for publication by The Scientific Concil of the Institute of Information Theories and Applications FOI ITHEA

This book maintains articles on actual problems of classification, data mining and forecasting as well as natural language processing:

- new approaches, models, algorithms and methods for classification, forecasting and clusterisation. Classification of non complete and noise data;
- discrete optimization in logic recognition algorithms construction, complexity, asymptotically optimal algorithms, mixed-integer problem of minimization of empirical risk, multi-objective linear integer programming problems;
- questions of complexity of some discrete optimization tasks and corresponding tasks of data analysis and pattern recognition;
- the algebraic approach for pattern recognition - problems of correct classification algorithms construction, logical correctors and resolvability of challenges of classification, construction of optimum algebraic correctors over sets of algorithms of computation of estimations, conditions of correct algorithms existence;
- regressions, restoring of dependences according to training sampling, parametrical approach for piecewise linear dependences restoration, and nonparametric regressions based on collective solution on set of tasks of recognition;
- multi-agent systems in knowledge discovery, collective evolutionary systems, advantages and disadvantages of synthetic data mining methods, intelligent search agent model realizing information extraction on ontological model of data mining methods;
- methods of search of logic regularities sets of classes and extraction of optimal subsets, construction of convex combination of associated predictors that minimizes mean error;
- algorithmic constructions in a model of recognizing the nearest neighbors in binary data sets, discrete isoperimetry problem solutions, logic-combinatorial scheme in high-throughput gene expression data;
- researches in area of neural network classifiers, and applications in finance field;
- text mining, automatic classification of scientific papers, information extraction from natural language texts, semantic text analysis, natural language processing.

It is represented that book articles will be interesting as experts in the field of classifying, data mining and forecasting, and to practical users from medicine, sociology, economy, chemistry, biology, and other areas.

General Sponsor: Consortium FOI Bulgaria (www.foibg.com).

Printed in Bulgaria

ISBN 978-954-16-0042-9

C\o Jusautor, Sofia, 2010

COMPUTER SUPPORT OF SEMANTIC TEXT ANALYSIS OF A TECHNICAL SPECIFICATION ON DESIGNING SOFTWARE

Alla V. Zaboleeva-Zotova, Yulia A. Orlova

Abstract: *The given work is devoted to development of the computer-aided system of semantic text analysis of a technical specification. The purpose of this work is to increase efficiency of software engineering based on automation of semantic text analysis of a technical specification. In work it is offered and investigated a technique of the text analysis of a technical specification is submitted, the expanded fuzzy attribute grammar of a technical specification, intended for formalization of limited Russian language is constructed with the purpose of analysis of offers of text of a technical specification, style features of the technical specification as class of documents, , algorithmic support of semantic text analysis of a technical specification and construction of software models are considered, recommendations on preparation of text of a technical specification for the automated processing are formulated. The computer-aided system of semantic text analysis of a technical specification is considered. This system consist of the following subsystems: preliminary text processing, the syntactic and semantic analysis and construction of software models, storage of documents and interface.*

Keywords*: natural language, semantic text analysis, technical specification.*

ACM Classification Keywords*: I.2.7 Natural Language Processing*

Introduction

Most known of the commercial software products used at designing of the software, basically are intended for visualization intermediate and end results of process of designing. Some of them allow to fully automate last design stages: generation of a code, creation of the accounting and accompanying documentation, etc. Thus the problem of automation of the initial stage of designing - formations and the analysis of the text of the technical project remains open. It is connected to extraordinary complexity of a problem of synthesis and the analysis of semantics of the technical text for which decision it is necessary to use methods of an artificial intellect, applied linguistics, psychology, etc. However, it is possible to come nearer to achievement of the given purpose, having allocated some small subtasks quite accessible to the decision by known methods of translation.

Proceeding from the aforesaid, it is possible to draw a conclusion, that the problem of creation of means for automation of process of designing is actual [1]. Ideas of a developed direction realization of the unified procedures of the designing equally answering to requirements of the expert - designer and requirements to technology to modelling of software products is main.

Area knowledge of software design consists of the following topics: basic concepts of design software, the key issues of designing software structure and software architecture, analysis and evaluation of the quality of design software, notation software design, strategy and methods of designing software.

In the modern information technology has an important place tools, systems development and maintenance of software. These technologies and the environment form a CASE-systems. The well-known CASE-systems, such as BPWin, ERWin, OOWin, Design / IDEF, CASE-Analyst, Silverrun, Rational Rose, Vantage Team Builder, S-Designer, etc., allow partially automate the process of designing software. However, as shown by the analysis, these systems automate the final stages of design software, such as the creation of the balance sheet and accompanying documentation, code generation, etc. The initial phase of the design - text analysis specification and construct a model of software - runs an analyst and automate of this phase remains open.

On CAD-department of the Volgograd state technical university questions of automation of designing of software products with use of natural - language support for a number of years are investigated.

Ideas of a developed direction realization of the unified procedures of the designing equally answering to requirements of the expert - designer and requirements to technology to modelling of software products is main.

Designing of the software at the initial stages with use of a natural language is based on the following main principles:

1. Performance of all design procedures is modelled in language of internal representation of system. Internal representation is the unified model of designing of the software, based on methodology of the theory of systems and technologies of natural language processing.

2. A number of representations of the project is generated. Translation of a condition of the project into the certain language which is distinct from language of internal representation refers to as representation. Programming languages, natural languages or artificial formal languages of modelling of processes of designing can be attributed to such languages (UML, IDEF-diagrams, model of diagrams of streams of the data). Different representations reflect only separate aspects of the project.

3. Thus due to use of uniform internal model consistency of representations is provided.

4. The software of process of the designing, guaranteeing an opportunity of conducting the project on any of languages of representations is developed.

5 The basic language of representation of the project for the person - the customer and the designer - is the natural language. Dialogue between the customer and the designer is traditionally conducted in a natural language - language of human dialogue, but, as a rule, are entered new formalism- diagrams, circuits, schedules. According to the developed concept, natural - language representation of the project supplements formal and serves as the tool facilitating understanding of process of designing.

As illustration of process of designing ON with use of the offered concept the diagram «to be», resulted on figure 1 serves.

The given work is devoted to development of the computer-aided system of semantic text analysis of a technical specification.

The purpose of this work is to increase efficiency of software engineering based on automation of semantic text analysis of a technical specification (TS). To achieve this purpose it is necessary to solve the following tasks:

1. To carry out the analysis of software engineering process and models of semantic text analysis;

2. To develop a technique of the text analysis of a technical specification;

3. To develop and investigate semantic model of the text of a technical specification;

4. To develop algorithmic maintenance of analysis of text of a technical specification and automatic construction of the software models;

5. To realize developed formalisms, a technique and algorithms as system of automation of the initial stage of designing software.

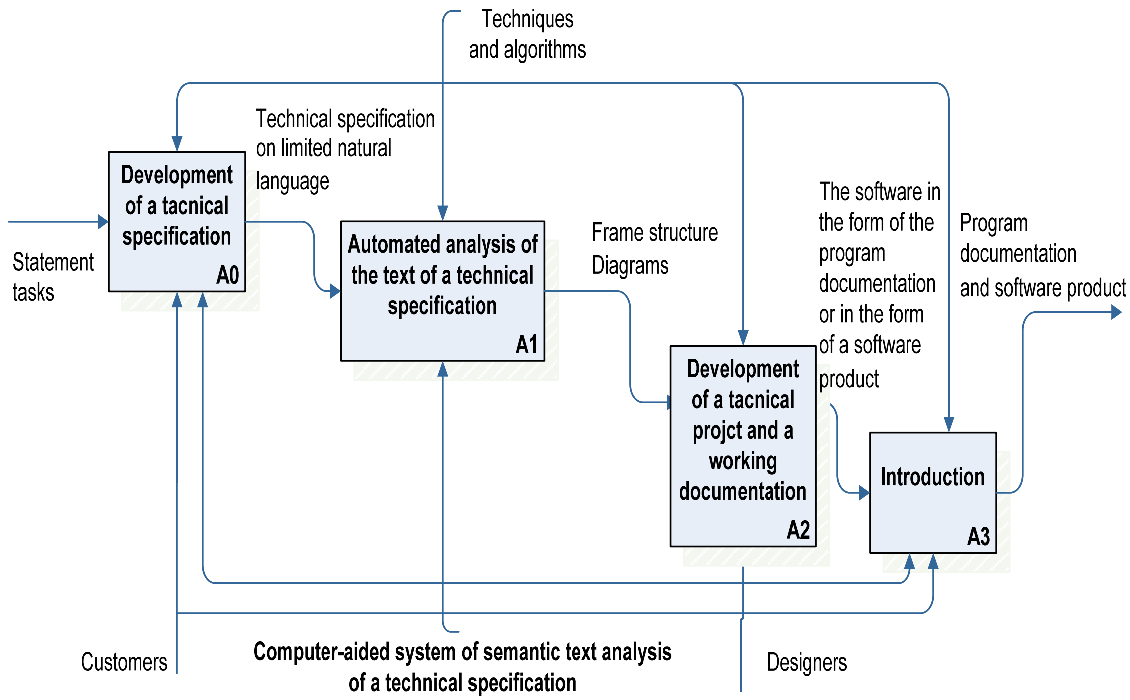


Figure 1: Diagram of process of designing «TO BE»

A Technique Of The Text Analysis Of A Technical Specification

In work it is offered and investigated a technique of the analysis of the text of a technical specification is submitted, the fuzzy attribute grammar of a technical specification, intended for formalization of limited Russian is constructed with the purpose of analysis of offers of text of a technical specification, style features of the technical specification as class of documents are considered, recommendations on preparation of text of a technical specification for the automated processing are formulated.

A technique of the analysis of the text of a technical specification consist of three stages: semantic text processing, creation of frame structure and creation of data flow diagrams of system described in the technical specification. (see Figure 2).

For realization of the first stage of a technique the semantic model of the text of a technical specification, including the requirements formulated as the document in the limited natural language has been developed; the second stage - the frame structure being internal representation of requirements; the third stage - model of software as the description of requirements in graphic language Data Flow Diagrams.

The semantic model of the text of a technical specification contains the developed expanded fuzzy attribute grammar above frame structure of the formal document "Technical specification" which allows to display contents TS most full.

The expanded fuzzy attribute grammar, necessary for the automated analysis of the text of a technical specification, is determined as:

$$AG = \langle N, T, P, S, B, F, A, D(A) \rangle,$$

where N - final set of non-terminal symbols; T - not crossed with N set of terminal symbols; P - final set of rules; S - the allocated symbol from N, named an initial symbol; B - set of linguistic variables $\beta_{k,i}$, corresponding to terminal symbols T (a variable i on k level); F - set of functions of a belonging $f_{k,i}$, determining a degree of belonging $m_{k,i}$

linguistic variables $\beta_{k,i}$; A - set of attributes, A = AsinUAsem, where Asin - syntactic attributes, Asem - semantic attributes; D (A) - final set of semantic actions. The fragment of grammar is submitted in table 1.

Linguistic variables from set B = $\{\beta_{k,i}\}_{k,i}$ used for the analysis of the text of a technical specification is described by the following five:

$$\beta_{k,i} = <\beta, T(\beta), U, G, M>,$$

β - name of linguistic variable (basis for development, purpose of development, technical requirements to a program product, a stage and development cycles, etc.);

T(β) - language expressions. For linguistic variables of the top level they are the linguistic variables corresponding to terminals of the right part of a rule. For linguistic variables of the bottom level – fuzzy variables, that is expressions of a natural language.

U - Set of all probable values, T (β) $\subset$ U;

G - rules of the morphological and syntactic description of language expressions which determine syntactic attributes Asin;

M - a semantic rule for linguistic variables which is induced by morphological and syntactic rules as the sense of a term in T is in part determined by its syntactic tree, and semantic attributes Asem.

Methods of representation connections between rules are broadcast on language of fuzzy mathematics. Thus connections are represented by fuzzy relations, predicates and rules, and sequence of transformations of these relations - as process of an fuzzy conclusion.

Linguistic variables of the top level are compound, that is include linguistic variables of the bottom level. Due to this it is possible to construct a tree of linguistic variables and to establish dependence between them.

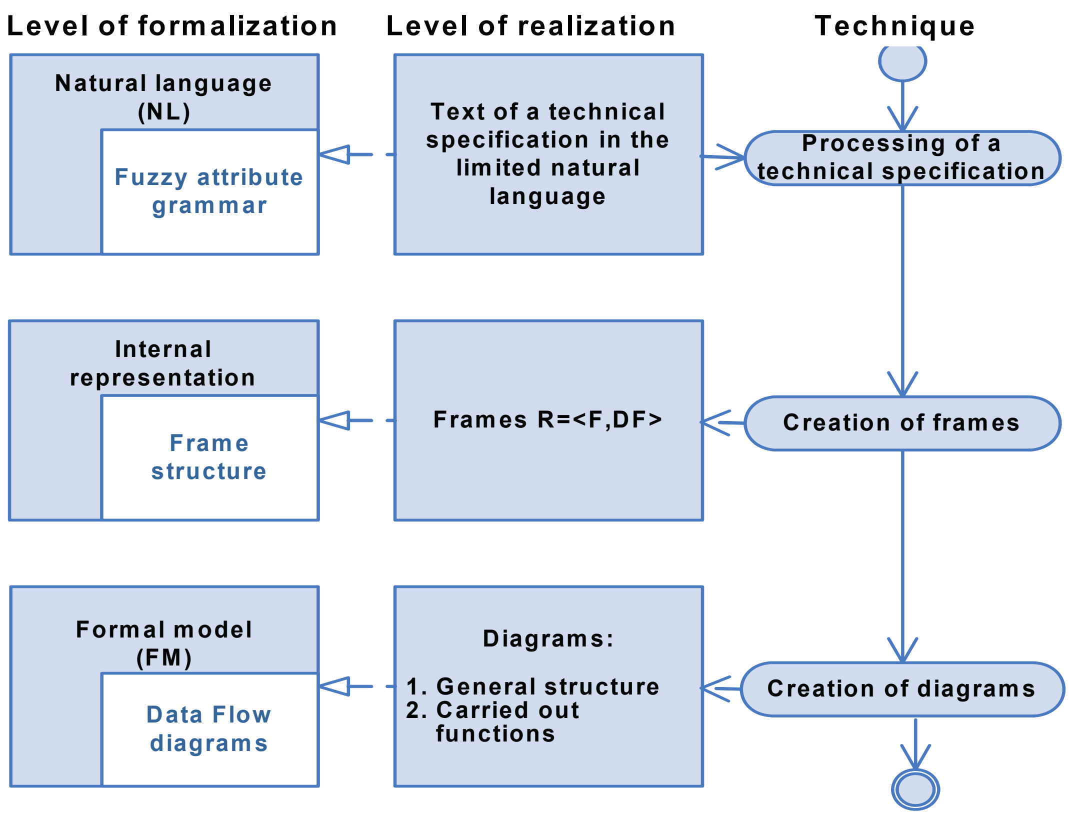


Figure 2: Technique Of The Text Analysis Of A Technical Specification

Table 1: Fragment of the developed fuzzy attribute grammar above frame structure of a technical specification

β_1	<*list of incoming data flows* >	<incoming data flow name > **:: 'Name'** <incoming data flow description> **:: 'Contents'** < list of incoming data flows >| ε
	<*incoming data flow description*>	The text containing "entrance" or "entrance data "**:: 'Clause**' <incoming data flow>**::"Frame Data Flow**=Creation **", "Input=Giving"**
$\beta_{1,2}$	<*incoming data flow*>	[<*Number of data units*>]:: "Slot AMOUNT OF DATA = Giving" [<*Type of data*>]:: " SLOT TYPE OF DATA = Giving " <*the Name of incoming data flow* >:: " Slot NAME OF INCOMING DATA FLOW= Giving"
β_2	<*function specification* >	< function type < name of the functions liss > **:: 'Name'**< function description >**:: "Frame FUNCTION = Creation** "; < List of functions> | ε
$\beta_{2,1}$	< *function type* >	«main» | «basic» | «additional»
$\beta_{2,2}$	<*function description* >	< Name of function>**:: 'Name', " Slot NAME OF FUNCTION = Giving "** < List of incoming data flow > <List of out coming data flow>

Functions of an a belonging from set F = $\{f_{k,i}\}_{k,i}$ linguistic variables $\{\beta_{k,i}\}_{k,i}$, are necessary for construction of an fuzzy conclusion. In particular, to each rule of grammar from set P function of a belonging $f_{k,i}$ is put in conformity. This dual system of substitutions is used for calculation of sense of a linguistic variable.

Actually grammar of a technical specification is used for splitting the initial text of the document into sections and processings of most important of them for our problem. It needs precise observance of structure of the document. Technical specification represents the structured text consisting of sequence of preset sections.

The frame structure of the technical specification is submitted as:

$$R = \left\langle N_R, \overline{F_R}, \overline{I_R}, \overline{O_R} \right\rangle$$

where N_R is a name of system, F_R is system functions vector, I_R is incoming data flows vector, O_R is outgoing data flows vector.

$$\overline{F}_R = \left\langle F_R^1, F_R^2, ..., F_R^k \right\rangle \text{, then } F_R^i = \left\langle N_F^i, \overline{I}_F^i, D_F^i, G_F^i, H_F^i, \overline{O}_F^i \right\rangle,$$

Where N_F^i - a name of function F_R^i, $\overline{I}_F^i$ - incoming data flows vector of F function, D_F^i - the name of the action which are carried out by function, G_F^i - subject of the function action, H_F^i - restrictions on function, $\overline{O}_F^i$ - a outgoing data flows vector of F function.

Let's denote the data flow by DF (Data Flow), then I_R, O_R, I_F, O_F are denoted by:

$$DF = \left\langle N_{DF}, D_{DF}, T_{DF}, C_{DF} \right\rangle$$

Where N_{DF} - data flow name, D_{DF} - data flow direction, T_{DF} - data type in flow, C_{DF} - data units per frame.

The model proposed is represented as a frame model with "a-kind-of" links (see Figure 3).

Algorithmic support Of Semantic Text Analysis Of A Technical Specification And Construction Of Software Models

General algorithm of semantic text analysis of a technical specification consists of the following blocks: preliminary text processing, syntactic and semantic analysis and construction of software models. Preliminary text processing is carried out using the apparatus of finite state machine, one of which is shown in Figure 4.

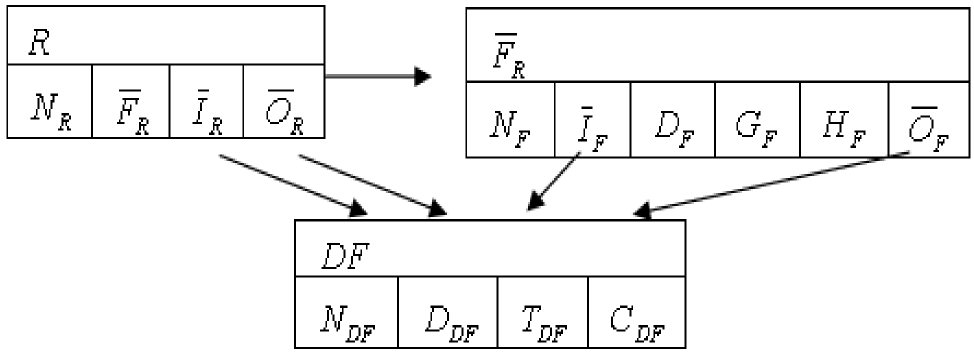


Figure 3: Frame network

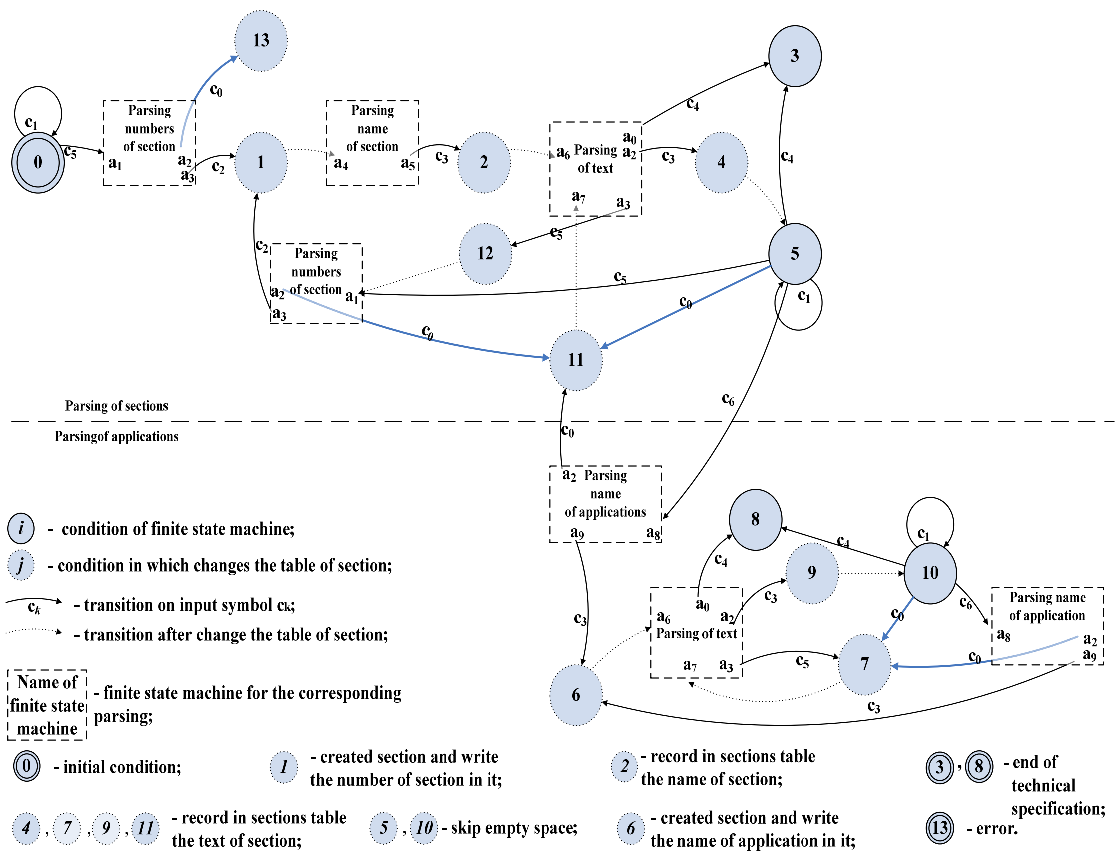


Figure 4: Finite state machine parsing of the text top-level technical specification

Preliminary text processing is necessary to share of a technical specification on separate lexemes. The incoming information of a subsystem is the text of a technical specification in the limited natural language, the target information - tables of sections, sentences and lexemes of a considered technical specification. Results can be submitted both as corresponding tables, and as a tree of sections.

Already after the first stage work not with the text of a technical specification, but with its parts submitted on sections is made. On a course of work of a technical specification shares all over again on more and more fine sections, then on separate sentences (with preservation of sections structure) and lexemes with the instruction of an accessory to sentences.

The input symbols of a finite state machine: c_1 - empty space, c_2 - space, c_3 - a new line, c_4 - the end of the text, c5 - '1 '.. '9', c6 - 'П', c0 - any other lexemes. Intermediate condition of finite state machine: a1 - start parsing section number, a_2 - a sequence of lexemes - text, a_3 - a sequence of lexemes - numbering, a_4 - start parsing the section name, a_5 - a sequence of lexemes - the section name, a_6 - start parsing the text of section or an application, a_7 - a sequence of lexemes - the continuation of the text section or application, a_8 - start parsing the application name, a_9 - a sequence of lexemes - the name of the application, a_0 - the end of technical specification.

In the course of a finite state machine lexemes acting on its entrance, collect in the buffer. In certain conditions, finite state machine the recording of the current contents of the buffer in one of the tables, after which the buffer is emptied. Work of finite state machine proceeds up to achievement of a final condition.

The output of the algorithm preliminary text processing of the text formed a set of tables: sections, sentences and lexemes. After this table obtained are fed to the algorithm of semantic analysis (Fig. 5).

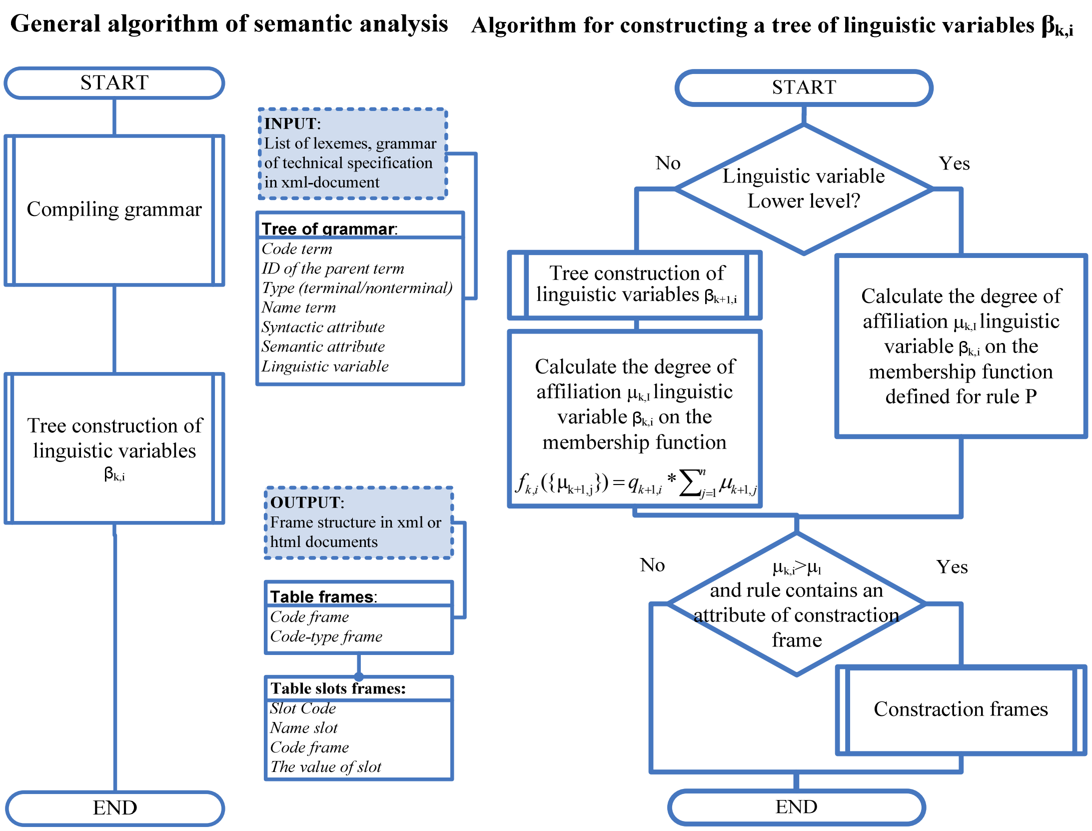


Fig. 5. Algorithm for semantic text analysis of a technical specification

The semantic analysis of a text is made on the basis of the developed grammar of text of a technical specification.

Rules of top level serve for analysis of sections of top level. Rules for analysis of sections consist of two parts: the first part serves for analysis of a section name; the second part serves for analysis of a text contents in section. Symbols of the given grammar possess syntactic attributes. In attributes of non-terminal symbols names of frames or names of slots in which the information received during the further analysis should be placed are specified. Syntactic attributes of text can be in addition specified in attributes of terminal symbols. Comparison of words at analysis is made in view of their morphology. During analysis the syntactic and morphological analysis are made only in the event that there is such necessity that time of performance of semantic analysis is considerably reduced.

Let's consider a fragment of the developed attribute grammar submitted in a xml-format:

… <global-rule id="Section42" comment = "Section 4.2. Requirements to functional characteristics"> <rule><ruleref uri="#Section42Name"/><ruleref uri="#Section42x"/></rule></global-rule>

<global-rule id="Section42Name" sectionPart="Name" comment= "Heading of the unit 4.2."><rule><clause clauseType="UNCERTAIN"/><rule type="or"><words contains="Functions"/> <words contains= " functional characteristics "/> </rule></rule></global-rule>

<global-rule id="Section42x" frame= "FunctionFrame" frameSlot="Function" comment="Function"><rule> <ruleref uri="#Section42xName" /><ruleref uri="#Section42xContent" /> </rule></global-rule>

<global-rule id="Section42xContent" sectionPart="Content" comment="Inputs and outputs of function"><rule><ruleref uri= "#Section42xInputs" minOccurs="0"/><ruleref uri="#Section42xOutputs" minOccurs="0"/></rule></global-rule>

<global-rule id="Section42xInputs" comment="Inputs of function">

<rule><sentence/><clause/><rule type="or"><words contains="Inputs"/> <words contains="entrance data"/></rule><ruleref uri="#Input" maxOccurs="unbounded"/></rule></global-rule> …

Semantic analysis is based on the developed fuzzy attribute grammar over the frame structure of text of a TS:

1. Each linguistic variable of a technical specification being reviewed, to result in the linguistic tree, end-vertices are fuzzy variables.

2. Fuzzy variables in the final vertices of the tree is assigned their meaning and then using a system of rules P and the corresponding membership functions $f_{k,i}$ is determined by the meaning of the linguistic variable corresponding to the left side of the rule.

Rules of the upper levels are used to parse sections of the upper level. The rules for parsing section consists of two parts: the first part is to parse the title of the section, the second part is to parse the text content section.

For some linguistic variable $\beta_{k,i}$ value of membership function: $\mu_{k,i}= f_{k,l} (\mu_{k+1,1}, \mu_{k+1,2}, \ldots, \mu_{k+1,n})$, where the specific value $\mu_{k,i}$ - degree of linguistic affiliation variable $\beta_{k,i}$. Initially, we say that all the linguistic variables of the lower level make the same contribution to the value of membership function, so you can say that the membership function of linguistic variable $\beta_{k,i}$:

$$f_{k,i}(\{\mu_{k+1,j}\}) = q_{k+1,i} * \sum_{j=1}^{n} \mu_{k+1,j}$$

where $\mu_{k+1,j}$ - degree of linguistic affiliation variable $\beta_{k,i}$; $q_{k+1,j} = 1 / n$ - contribution of degrees of linguistic affiliation variables in the value of membership function. At the lower level membership function are defined.

Calculated $\mu_{k,i}$ compared with μ_l, which is the limiting value of the degree of affiliation. If $\mu_{k,i} > \mu_l$, and the rules specified syntactic or semantic attributes, then creates frames and slots, which can hold the text of the linguistic variable.

3. Then the tree of linguistic variables cutting back so calculated linguistic variables were end-vertices of the remaining subtree.

This process is repeated until there is no sense to calculate the linguistic variable corresponding to the root of the source tree. The main purpose of this procedure is to associate the meaning of a linguistic variable with the meaning of its fuzzy variables by fuzzy attribute grammar above frame structure of a technical specification.

During parsing syntactic and morphological analysis is only done if there is a need, which significantly reduces the run-time semantic analysis. If the rules of grammar meets terminal with syntactic attribute, then run the parsing mechanism of semantic analysis for the current sentences [2]. After creating a tree of linguistic variables begin construction framing descriptions of a technical specification. It uses information about the frames and the names of slots, which is contained in the attributes of grammar symbols.

The resulting frame structure contains significant information about the system: information about the inputs and outputs of the system, functions and limitations. For each function is also provided inputs and outputs. This allows on the basis of frame structure to obtain a Data Flow Diagrams, which is described in the technical specifications. Algorithms for creating frames carries out construction and ordering the column of data flows, and also creation the figures of data flow diagrams in Microsoft Office Visio (Fig. 6).

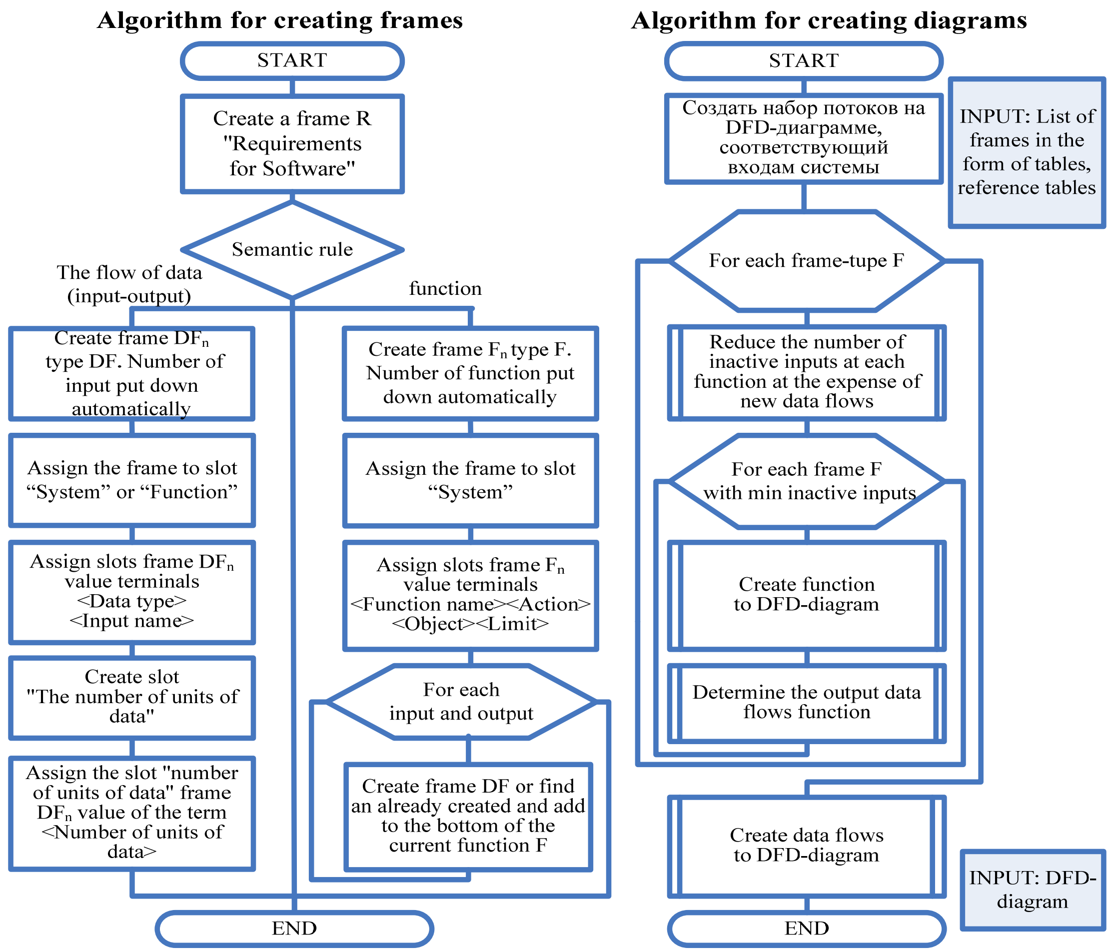


Fig. 6. Algorithms for creating frames and construction of Data Flow Diagrams

For construction of data flows it is prospected of functions inputs conterminous to system inputs. Then functions on which all inputs data act, are located on the one level of diagram. Their inputs incorporate to system inputs. Further it is prospected functions which inputs coincide with outputs of functions received on the previous step. They are located on the following level, their inputs incorporate to outputs of the previous levels functions and with system inputs.

Work of algorithm proceeds until all functions will not be placed on the diagram. After that connection of function outputs with necessary system outputs is made.

Computer-Aided System Of Semantic Text Analysis Of A Technical Specification

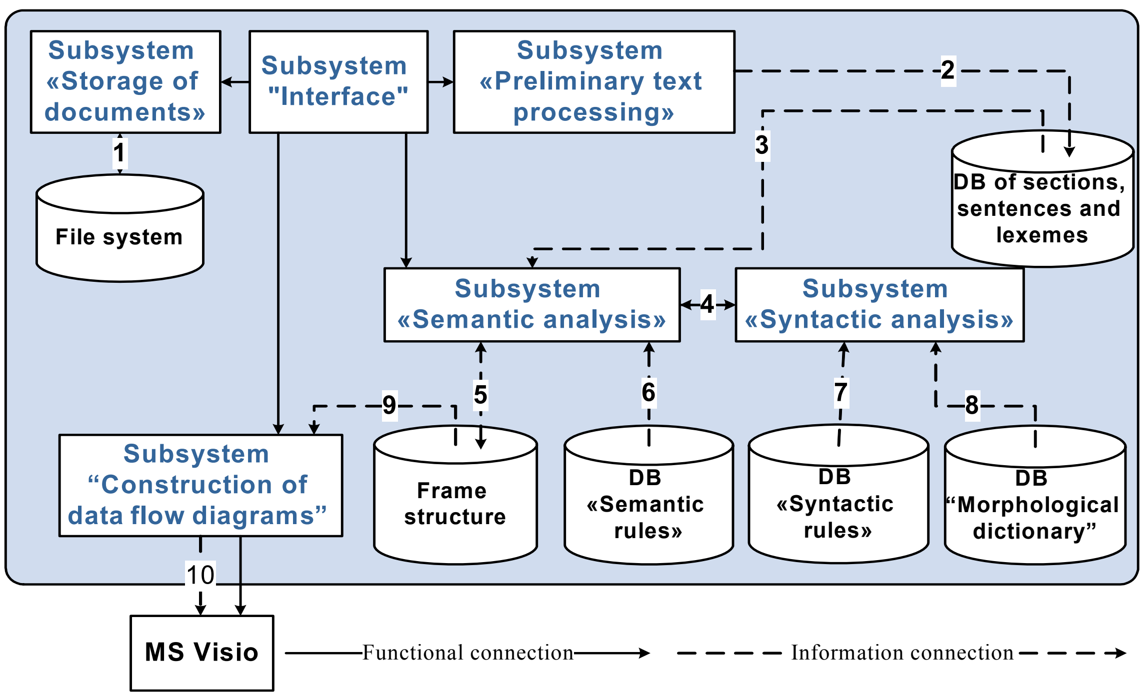
The computer-aided system of semantic text analysis of a technical specification consist of the following subsystems: preliminary text processing, the syntactic and semantic analysis and construction of software models, storage of documents and interface (see Figure 7).

The computer-aided system of semantic text analysis of a technical specification is developed on Microsoft .NET Framework 2.0 platform (language of development C#) using integrated development environment Visual Studio 2005. Tables are stored in XML, and their visual representation is possible using XSL-transformation. Obtained in the semantic analysis of the framing description is also stored in XML. Building a data flow diagram by means of interaction with the program MS Visio.

Scientific Novelty

Scientific novelty consists in the following: a technique of text analysis of a technical specification at the initial stages of software engineering, including semantic model of text of a technical specification, transformation matter of text into the frame structure and construction of model of the software on its basis are developed.

1. New semantic model of text of a technical specification is represented as a fuzzy extended attribute grammar over frame structure, containing syntactic, semantic attributes and linguistic variables.



1 - Files of documents
2, 3 - Data about sections, sentences and lexemes
4 - Syntactic units, groups, clause
5, 9 – Frames
6 - Semantic rules
7 - Syntactic rules
8 - Morphological attributes
10 - Inscriptions and coordinates of data flow diagrams figures

Figure 7: Architecture of computer-aided system of semantic text analysis of a technical specification

2. Proposed and developed methods and algorithms designed to transform the source text of a technical specification developed in a frame structure, which is a description of the requirements to the software.

3. A method of constructing models of the software described in the technical specification in the form of data flow diagram is developed.

Practical Value

Practical value of work is that as a result of development and introduction of a suggested technique quality of software engineering raises due to automation of routine work of the person on extraction of helpful information from standard documents and to displaying it as software models. Thus, developed an automated system improves to increase efficiency of software design at the initial stage by reducing the time working on technical specifications and increase the quality of the result. Software designing differs from designing in other areas of a science and technics a little, therefore it is possible to expand results of the given work for application in other areas of human knowledge. Thus, opening prospects raise a urgency of the given work.

Conclusions and Future Work

The result of this work the following results:

1. The analysis of the process of designing software, the existing models and methods for word processing is carried out, show the importance of the specification of projects and the importance of the analysis stage in the process of software development. Justified necessity of automating the initial stages of software design and, in particular, the semantic analysis of text specification to identify the functional structure of the system described in the specifications.

2. Developed technique of text analysis specification, designed for text processing in the early stages of software design and containing the formalisms necessary for representing the semantics of the software requirements at the early stages: the semantic model of the text specification, frame structure and a formal model. The technique of analysis involves three stages: semantic processing of text, creating frame structure and the creation of a model of software as a data flow diagram of the system described in the specifications.

3. Analyzed the stylistic features of the text specification, based on which developed a semantic model of the text specification, is an extended fuzzy attribute grammar over a frame structure containing syntactic, semantic attributes and linguistic variables. A frame structure, which is an internal representation requirements to the software and allows the automated system to be universal with respect to the natural language user-designer.

4. Proposed algorithms of semantic analysis of text and technical specification: a preliminary text processing, syntactic and semantic analysis and modeling software. preliminary text processing by using the apparatus of finite state mashine, which results are generated tables of section, sentences and lexemes. Semantic analysis of text is based on the developed fuzzy attribute grammar. Developed algorithmic operations to calculate the meaning of linguistic variable and the construction of fuzzy inference. Based on the tree of linguistic variables and semantic attributes created frame description of the system and implemented the construction of a model of software as a data flow diagram.

5. Developed formalisms, methods and algorithms are implemented in the form of automation systems initial stage of software design "SemantikaTS" platform Microsoft. NET Framework 2.0 (development language C #) using a visual programming environment Visual Studio 2005. Building a data flow diagram implemented in MS Visio.

Bibliography

Tools Development For Computer Aided Software Engineering Based On Technical Specification`s Text Analysis / A.Zaboleeva-Zotova, Y.Orlova // Interactive Systems And Technologies: The Problems Of Human-Computer Interaction: Proc. of the Int. Conf., Ulyanovsk.

Zaboleeva-Zotova, A. Computer Support of Symantic Text Analysis of a Technical Specification on Designing Software / A.Zaboleeva-Zotova, Y.Orlova // Intelligent Processing : suppl. to Int. Journal "Information Technologies and Knowledge" Vol. 3. - 2009. - Int. Book Series "Information Science & Computing", № 9. - C. 29-35.

Zaboleeva-Zotova, A. Formalization of text analysis of a technical specification / A.Zaboleeva-Zotova, Y.Orlova // Congress on intelligent systems and information technologies (AIS-IT`09), Divnomorskoe, Russia, September, 3-10 : proc. - M., 2009. - Vol. 4. - C. 62

Zaboleeva-Zotova, A. Automation of procedures of the semantic text analysis of a technical specification / A.Zaboleeva-Zotova, Y.Orlova // Congress on intelligent systems (INTELS`2008). - M., 2008. - P. 245-248.

Zaboleeva-Zotova, A. Computer-aided System of Semantic Text Analysis of a Technical Specification / A.Zaboleeva-Zotova, Y.Orlova // Advanced Research in Artificial Intelligence: suppl. to Int. Journal "Information Technologies and Knowledge". - 2008. - Vol. 2, [Int. Book Series "Inform. Science & Comput."; № 2]. - P. 139-145.

Orlova, Y Computer-aided system of the semantic text analysis of a technical specification / Y.Orlova // Distance Learning - Learning Environment of the XXI century: VI Intern. science method. Conf., Minsk, 22-23 November 2007 / Belorus. state. Univ of Informatics and Radioelectronics. - Minsk, 2007. - P.127-129.

Panchenko, D. Implementation of Genetic Algorithms for Transit Points Arrangement / D. Panchenko, M. Scherbakov / / Information Technologies & Knowledge: suppl. "Information Science and Computing". No. 9. - 2009. - Vol. 3. - C. 129-131 .- Engl.

Kamaev V., Shcherbakov, M., Skorobogatchenko D. Automated prediction of transport operating conditions of the roads. Journal of Computer and Information Technologies, № 4, 2004, Moscow: "Engineering", pp 2-6.

Kamaev V., Shcherbakov M. A neural approach, identification of complex systems, Journal of Computer and Information Technologies, № 3, 2004, Moscow: Mashinostroenie, S. 20-24.

Rosaliev, V. The model of emotion in human speech / V/ Rosalia / / Proceedings VolgGTU. A series of "Actual problems of management, computing and informatics in technical systems": Hi. Sat scientific. Art. / VolgGTU. - Volgograd, 2007. - Vol.3, № 9. - P.62-65.

Rosalie, V. Background, and prospects of creating an automated pattern recognition system emotional speech / V. Rosalia / / Proceedings VolgGTU. A series of "Actual problems of management, computing and informatics in technical systems": Hi. Sat scientific. Art. / VolgGTU. - Volgograd, 2008. - Issue 4, № 2. - P.58-61.

Authors' Information

Alla V. Zaboleeva-Zotova *– PhD, professor; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: zabzot@vstu.ru*

Major Fields of Scientific Research:

Yulia A. Orlova *– PhD; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: yulia.orlova@gmail.com*

Major Fields of Scientific Research: