Applying Multisets for Software Design at the Initial Stages

Yulia A. Orlova
CAD department
Volgograd State Technical University
Volgograd, Russia
e-mail: yulia.orlova@gmail.com

Alexey B. Petrovsky
Institute for Systems Analysis
Russian Academy of Sciences
Moscow, Russia
e-mail: pab@isa.ru

***Abstract*—The paper describes how to apply multisets for software design. Developed system for automation of the initial stages of multi-component software design is based on the semantic analysis of text. Automation lead to increasing in the quality of software development. The paper considers an approach to automation of the initial stages of software design. Program tools, which provide automated semantic analysis of technical documentation, automated construction of models, synthesis of structure and natural language description of the program software, are developed.**

Keywords-software design,multisets,semantic text analysis

I. INTRODUCTION

Most of the known commercial software products, designed for a visualization of software systems' development, are used usually on the intermediate and final stages of the engineering process. The well-known CASE-systems such as BPWin, ERWin, OOWin, Design / IDEF, CASE-Analyst, Silverrun, Rational Rose, Vantage Team Builder, S-Designor and others, can partially automate the process of software design. For instance, some of these products automate fully a generation of program code, a creation of the accounting and accompanying documentation, etc. But an automation of initial stages of software design is still an open problem that can be solved by using methods of artificial intelligence, applied linguistics, and so on.

Important in the software development process are also means the specification of software projects. In the initial stages of software design is based functional model of system that describes a set of functions. The construction of functional specifications is performed by the analyst based on the text of technical documents and is not currently an automated procedure.

As special study, 41% of errors found in software project are mistakes made in the initial stages of design. The reason for 56% of all errors are errors made during the stage of requirements analysis [1]. Such errors are extremely costly to correct.

The analysis leads to the following conclusions:

• Automated largely separate stages of design.

• There are no methods for automated analysis of technical documentation.

• There is no automatic process modeling software.

• In the initial stages of software design may be a lot of mistakes.

Thus, the automation of the initial stages of software design remains an open problem, which requires for its solution the creation of special software.

The purpose of this work is to increase efficiency of software designing based on automation semantic text analysis of a technical specification. It is necessary to solve the following tasks: to develop a technique of the semantic text analysis of a technical specification; to develop algorithmic maintenance of analysis of text of a technical specification and automatic construction of the software models; to realize developed formalisms, a technique and algorithms as system for automation of the initial stage of designing software.

In this paper, we consider the method and algorithms of the semantic text analysis, which are used for an analysis of a technical specification of multi-component software at the initial stages of software design. The computer-aided system that realizes the developed formalisms is described.

II. TECHNIQUE FOR SEMANTIC TEXT ANALYSIS OF THE TECHNICAL SPECIFICATION

Technique for semantic analysis of the technical specification text consists of four phases: semantic word processing, creating frame structure, the creation of software models, as described in the technical documentation, the synthesis of the description text of the constructed models [2].

In order to realize the word processing, the semantic model of the technical specification text, including the requirements formulated as the formal document "Technical Specification" in the limited natural language, has been developed. The internal representation of requirements is given by the frame structure.

The semantic model of the technical specification text contains the expanded fuzzy attribute grammar over the frame structure of the technical specification. This grammar allows to display the most fully the contents of technical specification [3]. The model of software is constructed as the description of requirements in the graphic language Data Flow Diagrams.

The expanded fuzzy attribute grammar [4], suggested for the automated analysis of the technical specification text, is determined as:

$$AG = <N, T, P, S_0, B, F, A, D(A)> \qquad (1)$$

Where N is the final set of non-terminal symbols;

T is the set of terminal symbols not intersected with N;

P is the final set of rules; S_0 is the allocated symbol from N, which is named an initial symbol;

B is the set of linguistic variables $\beta_{k,i}$, corresponding to terminal symbols T (a variable i on k level);

F is the set of functions $f_{k,i}$ determining a degree of membership $m_{k,i}$ of linguistic variables $\beta_{k,i}$;

A=Asyn⋃Asem is the set of attributes, where Asyn – the syntactic attributes, Asem – the semantic attributes;

D(A) is the final set of semantic actions.

The linguistic variable of the set $B=\{\beta_{k,i}\}$, which is used for an analysis of the technical specification text, is described as follows:

$$\beta_{k,i} = <\beta, T(\beta), U, G, M> \quad (2)$$

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

T(β) is a language expressions;

U is the universal set of all probable values such as $T(\beta)\subset U$;

G is rules of the morphological and syntactic description of language expressions, which determine syntactic attributes Asyn;

M is a semantic rule for linguistic variables, which is induced by morphological and syntactic rules.

The linguistic variables of the bottom level are fuzzy variables expressed in a natural language. The linguistic variables of the top level correspond to terminals of the right part of a rule and include linguistic variables of the bottom level. So, it is possible to construct a tree of linguistic variables and establish dependence between them.

Connections between the rules are described by fuzzy relations, predicates and rules. A sequence of transformations of these relations is represented as the process of fuzzy conclusions. Fuzzy conclusions are constructed with the set $F=\{f_{k,i}\}$ of membership functions of the linguistic variables $\{\beta_{k,i}\}$.

III. Information Model

Consider the system described in the technical specification as a set of input actions, functions, and output actions [5]:

$$S = (N_S, F_S, \boldsymbol{I}_S, \boldsymbol{O}_S) \quad (3)$$

Where N_S – name of system,

F_S – set of system functions,

$\boldsymbol{I}_S$ – multiset of input actions,

$\boldsymbol{O}_S$ – multiset of output actions.

Input actions of such a system S are a multiset:

$$\boldsymbol{I}_S = \{k_I^1 \cdot I_S^1, \ldots, k_I^n \cdot I_S^n, \ldots, k_I^N \cdot I_S^N\} \quad (4)$$

Where function of frequency rate k_I^n characterizes number of identical input I_S^n.

Output actions of such a system S are a multiset:

$$\boldsymbol{O}_S = \{k_O^1 \cdot O_S^1, \ldots, k_O^m \cdot O_S^m, \ldots, k_O^M \cdot O_S^M\} \quad (5)$$

Where function of frequency rate k_I^n characterizes number of identical output O_S^m.

Set of system functions F_S are realized by subsystems $F_S^1,\ldots,F_S^l,\ldots,F_S^L$, each of which is described as follows:

$$F_S^l = (N_F^l, \boldsymbol{I}_F^l, D_F^l, G_F^l, H_F^l, \boldsymbol{O}_F^l) \quad (6)$$

Where N_F^l – name of function F_S^l,

$\boldsymbol{I}_F^l$ – multiset of input actions of function F_S,

D_F^l – name of action which is carried out by function,

G_F^l – subject of the function action,

H_F^l – restrictions on function F,

$\boldsymbol{O}_F^l$ – multiset of output actions of function F_S.

Multisets $\boldsymbol{I}_S$, $\boldsymbol{O}_S$, $\boldsymbol{I}_F^l$, $\boldsymbol{O}_F^l$ represent data flows which consist of the single portions having identical structure:

$$DF = (N_{DF}, D_{DF}, T_{DF}) \quad (7)$$

Where N_{DF} – data flow name,

D_{DF} – data flow direction,

T_{DF} – data type in flow.

Operations over multisets it is possible to use for synthesis of software structure [6], [7].

At synthesis of text of the constructed models description requirement to software are represented in the form of a set of data flows in the graphic Data Flow Diagrams language. In diagrams the general structure of system with the indication of its inputs, outputs and functions which are carried out by system is displayed.

IV. Algorithmic Support

The general algorithm for semantic analysis of the technical specification text consists of the following blocks: preliminary text processing; the syntactic and semantic analysis; a construction of software models [8].

Preliminary text processing is carried out using the apparatus of finite state machine in order to share the text of the technical specification on separate lexemes. The incoming information of the first block is the text of the technical specification in the limited natural language, the target information are tables of sections, sentences and lexemes of the considered technical specification. The obtained tables are fed to the algorithm of semantic analysis.

The semantic analysis of text is performed by using the developed grammar of the technical specification text. Symbols of the grammar possess the syntactic attributes. In attributes of the non-terminal symbols, the names of frames or slots, in which the information receiving during the further analysis should be placed, are specified. In attributes of the terminal symbols, the syntactic attributes can be specified additionally.

1. Each linguistic variable of a technical specification being reviewed, to result in the linguistic tree, end-vertices are fuzzy variables [9].

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 rule 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:

$$\mu_{k,i}=f_{k,I}(\mu_{k+1,1}, \mu_{k+1,2}, \ldots, \mu_{k+1,n}), \quad (8)$$

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 $\beta_{k,i}$:

$$f_{k,i}(\{\mu_{k+1,j}\})=q_{k+1,i}\bullet\Sigma^{n}_{j=1} \mu_{k+1,j} \quad (9)$$

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.

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

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.

The syntactic and morphological analysis is done during parsing only if there is a need to reduce significantly the run-time of semantic analysis. If the rules of grammar meets terminal with the syntactic attribute, then the parsing mechanism of semantic analysis runs for the current sentences. After creating a tree of linguistic variables, a frame description of the technical specification text is constructed on the basis of grammar containing information about frames and slots.

The structure of the resulting frame includes significant information about the inputs and outputs of the system, functions and limitations. For each function it is also provided inputs and outputs. Basing on frame structure, Data Flow Diagrams, which are described in the technical specifications, are obtained.

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.

V. Computer-aided System

The proposed formalisms, methods and algorithms are implemented in the computer-aided system "Semantika TS" developed for automation of the initial stages of multi-component software design. Automated system consists of six subsystems.

The subsystem "Storage of documents" provides a possibility of loading and saving text documents, tables of sections, sentences and lexemes, tables of frames, and reports on operations of the lexical and semantic analyzer in the XML format.

The subsystem "Interface" provides a management of sessions, visual representation of XML documents and reports.

The subsystem "Preliminary text processing" carries out loading structure of the final state machine for analysis of text, creation tables of sections, sentences and lexemes of the analyzed technical specification.

The subsystem "Syntactic analysis" carries out a search of the best syntactic option, formation of the list of syntactic units and syntactic groups.

The subsystem "Semantic analysis" loads the description of frame of data structure, forms a state of machine for semantic analysis and creation of a semantic tree of the technical specification.

The subsystem "Constructions of data flow diagrams" constructs a data flow graph and creates figures of data flow diagrams in Microsoft Office Visio.

The computer-aided system for semantic text analysis is developed on Microsoft.NET Framework 2.0 platform (language of development C#) using integrated development environment Visual Studio. Tables are stored in XML, and their visual representation is possible using XSL-transformation. Frame description is also stored in XML. Data flow diagrams are built with the program MS Visio.

Research of the developed computer-aided system at software design is conducted. Increase of efficiency consists in considerable reduction of time of text specification analysis and creation of model software (from 40% to 70% depending on number of functional units in technical specification). Also considerably the percent of detection and correction of the mistakes made at a stage of formalization and the analysis of technical specification increased.

VI. Conclusion

Scientific novelty: a technique of text analysis of a technical specification at the initial stages of software engineering, including semantic model, transformation matter of text into the frame structure and construction models of software 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.

2. Proposed and developed methods and algorithms designed to transform the source text of a technical specification developed in a frame structure.

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 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 a little, therefore it is possible to expand results of the given work for application in other areas of human knowledge.

Clearly indicate advantages, limitations and possible applications. In this paper, we presents the technique of text analysis of technical specification at the initial stages of software design, which includes the semantic model of a technical specification text, tools for transforming text into the frame structure, preliminary text processing, constructing data flow diagrams, and modeling the software. Development and application of this technique lead to increasing in the quality of software design due to automation of routine work of person to extract helpful information from standard documents and represent as the software models [10].

The automated system for multi-component software design at the initial stages can be used in organizations preparing the technical specifications of software as the legal document [11]. The suggested approach can be also useful to automate analyzing and preparing the specifications of various technical and information systems.

In future we are going to generate diagrams in UML language.

VII. Acknowledgement

This work is supported in part by the Russian Foundation for Basic Research 11-07-00398-a, 12-07-00266-a, 12-07-00270-a, 13-07-00351-a, 13-07-00459-a, 13-07-00461-a.

References

[1] A. V. Zaboleeva-Zotova and Yu. A. Orlova, "Automation of semantic text analysis of a technical specification". Monograph. Volgograd, IUNL, 2010 (in Russian).

[2] A. V. Zaboleeva-Zotova and Yu. A. Orlova, "Computer support of semantic text analysis of a technical specification on designing software". Intelligent Processing: Supplement to International Journal "IT and Knowledge", 2009, pp. 29-35.

[3] A. V. Zaboleeva-Zotova and Yu. A. Orlova, "Computer-aided system of semantic text analysis of a technical specification". Advanced Research in Artificial Intelligence: Supplement to International Journal "IT and Knowledge", 2008, pp. 139-145.

[4] V. Novak, I. Perfileva and I. Mochkorzh, "Mathematical Principles of Fuzzy Logic". Fizmatlit, Moscow, 2006 (in Russian).

[5] A. V. Zaboleeva-Zotova, Yu. A. Orlova and A. B Petrovsky, "Automation of Software Design on Initial Stages". Advances in Decision Technology and Intelligent Information Systems. Vol. XIII, 2012, pp. 26-30

[6] A. B. Petrovsky, "Kombinatorika of multisets". Reports of Academy of Sciences. Vol. 370, No. 6, 2000, pp. 750-753 (in Russian).

[7] A. B. Petrovsky, "Prostranstva of sets and multisets". Moscow. Editorial of URSS, 2003 (in Russian).

[8] U. Reyle, "Natural language parsing and linguistic theories". Rohrer Dordrecht, Berlin, 2008.

[9] A. E. Kibrik, "Constants and variables of language". Aletheia, Saint-Petersburg, Russia, 2003 (in Russian).

[10] Yu. Orlova and V.L., "Rozaliev Increase efficiency of software engineering by improving interaction with the user". Proceedings of the VI International Scientific-Practical Conference "Integrated models and soft computing in artificially intelligence". Moscow, Fizmatlit, 2011, pp. 907-914 (in Russian).

[11] I. Ashikhmin, E. Furems, A. Petrovsky and M. Sternin, "Intelligent DSS under verbal decision analysis". Encyclopedia of Decision Making and Decision Support Technologies, ed. by F.Adam, P.Humphreys. IGI Global, Hershey, New York, Vol.2, 2008, pp. 418-425 (in Russian).