KAUNAS UNIVERSITY OF TECHNOLOGY

Germanas Budnikas

DEVELOPMENT AND ANALYSIS OF AGGREGATE SPECIFICATIONS USING KNOWLEDGE BASES

Summary of Doctoral Dissertation

Physical sciences, Informatics (09P)

Kaunas, 2004

The research was accomplished during the period of 1999 to 2003 at Kaunas University of Technology, Business Informatics Department.

Academic supervisor:

Prof. Dr. Habil. Henrikas PRANEVIČIUS (Kaunas University of Technology, Physical Sciences, Informatics - 09P).

Council of Informatics trend:

Prof. Dr. Habil. Vytautas ŠTUIKYS (Kaunas University of Technology, Physical Sciences, Informatics - 09P) - **chairman**,

Prof. Dr. Habil. Rimantas BARAUSKAS (Kaunas University of Technology, Physical Sciences, Informatics - 09P),

Prof. Dr. Habil. Raimundas ČIEGIS (Vilnius Gediminas Technical University, Physical Sciences, Informatics - 09P),

Prof. Dr. Habil. Jonas MOCKUS (Institute of Mathematics and Informatics, Physical Sciences, Informatics - 09P),

Prof. Dr. Habil. Antanas VERIKAS (Kaunas University of Technology, Physical Sciences, Informatics - 09P).

Official opponents:

Assoc. Prof. Dr. Albertas ČAPLINSKAS (Institute of Mathematics and Informatics, Physical Sciences, Informatics - 09P),

Prof. Dr. Habil. Genadijus KULVIETIS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering - 07T).

The official defense of the dissertation will be held at 2 p.m. on December 22, 2004 at the Council of Informatics trend public session in the Dissertation Defense Hall at Kaunas University of Technology (K. Donelaičio g. 73, room No. 403, Kaunas).

Address: K. Donelaičio g. 73, LT-44029 Kaunas, Lithuania.

Tel.: +370 (37) 300042, fax: +370 (37) 324144, e-mail: mok.grupe@adm.ktu.lt

The send-out date of the summary of the Dissertation is on 22 of November, 2004.

The dissertation is available at the library of Kaunas University of Technology (K. Donelaičio g. 20, Kaunas).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Germanas Budnikas

AGREGATINIŲ SPECIFIKACIJŲ SUDARYMAS IR ANALIZĖ PANAUDOJANT ŽINIŲ BAZES

Daktaro disertacijos santrauka

Fiziniai mokslai, informatika (09P)

Kaunas, 2004

Disertacija rengta 1999 – 2003 metais Kauno technologijos universitete, Verslo informatikos katedroje.

Mokslinis vadovas:

Prof. habil. dr. Henrikas PRANEVIČIUS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P).

Informatikos mokslo krypties taryba:

Prof. habil. dr. Vytautas ŠTUIKYS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P) – **pirmininkas**,

Prof. habil. dr. Rimantas BARAUSKAS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P),

Prof. habil. dr. Raimundas ČIEGIS (Vilniaus Gedimino technikos universitetas, fiziniai mokslai, informatika – 09P),

Prof. habil. dr. Jonas MOCKUS (Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09P),

Prof. habil. dr. Antanas VERIKAS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P).

Oficialieji oponentai:

Doc. Dr. Albertas ČAPLINSKAS (Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09P),

Prof. habil. dr. Genadijus KULVIETIS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija bus ginama viešame Informatikos mokslo krypties tarybos posėdyje 2004 m. gruodžio 22 d. 14 val. Kauno technologijos universiteto Disertacijų gynimo salėje (K. Donelaičio g. 73, 403 a., Kaunas).

Adresas: K. Donelaičio g. 73, LT – 44029 Kaunas.

Tel.: (8-37) 300042, faksas: (8-37) 324144, el. paštas: mok.grupe@adm.ktu.lt

Disertacijos santrauka išsiųsta 2004 m. lapkričio 22 d.

Su disertacija galima susipažinti Kauno technologijos universiteto bibliotekoje (K. Donelaičio g. 20, Kaunas).

Object of the study. Aggregate specifications that are derived from informal descriptions of distributed systems.

Problem statement. Distributed systems become more complex, so do the architectures that underlie them. Software-based systems increasingly operate in changing environments under variable user requirements. One of the solutions for these problems is the use of formal methods during system design stage to develop and analyse models of such a system. Timed automata are a class of formal methods. Aggregate specification method belongs to this class. PRANAS tool-set supports application of formal aggregate specification method during the development of software systems—while creating and analysing aggregate specifications. The tool-set is developed in Business Informatics Department in Kaunas University of Technology. A tool-set component—a specification editor—automatically generates a framework of an aggregate specification, which consists of description of system modules in terms of states and rules for the state changes. The framework of the specification is heuristically supplemented with a non-validated description of algorithmic part of system functioning during the change of state.

It is very common to begin the development of the systems by giving domain-based systems description using only symbolical values. Knowledge-based systems and techniques support such an approach. Specification of the system can be given as a collection of knowledge about application domain, without giving precise numerical values. Initial description may be incomplete, ambivalent, and imprecise. Therefore, methods and techniques are indispensable to detect inconsistencies at the initial stage of system development, during the development of formal specifications, in this case an aggregate specification.

Aim of the study. The aim of the study is to develop technique for development and analysis of static and dynamic properties of aggregate specifications using knowledge bases. It can be decomposed to the following sub-tasks:

- Development of an aggregate specification knowledge base from an informal description;
- Static verification and dynamic validation of aggregate specification knowledge base;
- Development of an aggregate specification using validated and verified specification knowledge base;
- Experimentally proof of the proposed techniques by development and analysis of aggregate specifications of a queuing system with priorities, an alternating bit protocol, a network of queuing systems, and an Internet cache protocol.

Used methods and software tools:

- Piece linear aggregate specification method;
- Single hit decision table verification method and reachable states validation method;
- Aggregate specification editor PRAXIS was used for the generation of a specification framework;
- PROLOGA tool was used for static verification of knowledge bases;
- CLIPS tool was used for dynamic validation of knowledge bases;
- PRANAS tool-set was used for validation of aggregate specifications.

Presented for defending. A technique, which allows to create and to check in advance static and dynamic properties of the algorithmic part of an aggregate specification, using knowledge engineering means is presented:

1. Structure of the aggregate specification knowledge base oriented to the development of aggregate specifications and defined using production rules.

   The aggregate model is described in the knowledge base: input and output signals, operations, internal and external events, transition and output operators (changes of aggregate state), aggregate interconnection scheme.
2. Verification technique that checks static properties of the aggregate specification knowledge base.

   The technique is based on a single hit decision table representation, which is used to check the knowledge base properties, such as absence of redundancy, ambivalence and deficiency. This technique can be applied both for fragments of specification, which characterises reaction of the system to an event, and for a whole specification.
3. Validation technique that analyses dynamic properties of the aggregate specification knowledge base.

   The technique is based on a production rule representation and a forward inference engine. It allows checking such properties as reachability, state coordinate boundedness, absence of over specification and static deadlocks.
4. Created, validated and verified knowledge bases of the queuing system with priorities, the alternating bit protocol, the network of queuing systems and the Internet cache protocol.

   The proposed technique was applied for the development of aggregate specifications from the knowledge bases.

Scientific novelty of the work. A Knowledge engineering based technique to describe the algorithmic part of an aggregate specification was proposed and static properties during development stage were verified.

Approval of the research results

In Lithuanian conferences:

- Information technologies (Kaunas, 2000).

In international conferences and workshops:

- CONSA Special Workshop “Simulation Applications in the Baltic Area” (Riga, Latvia, 1999);
- Nordic and Northern Russia Summer School “Applied Computation Intelligence to Engineering and Business” (St. Petersburg, Russia, 2000);
- IFIP WG6.1 Joint International Conference - FORTE/PSTV 2000: Formal Techniques for Communication Protocols and Distributed Systems (FORTE XIII) and Protocol Specification, Testing and Verification (PSTV XX) (Pisa, Italy, 2000);
- CONSA Special Workshop “Simulation: Applications, Research and Education in the Baltic Area” (Linkoping, Sweden, 2001);
- International Conference “Modelling and Simulation of Business Systems” (Vilnius, 2003);
- Sixth International Baltic Conference “Databases & Information Systems 2004” (Riga, Latvia, 2004).

Publications. There are 7 scientific papers published on the topic of the dissertation:

- In editions included in the main list of Institute of Scientific Information 1;
- In editions included in the list approved by the Department of Science and Study 2;
- In other reviewed international and foreign editions 3;
- In proceedings of Lithuanian conferences 1.

Structure and size of the dissertation. The dissertation consists of: an introduction; six chapters (including conclusions); a list of references and publications; annexes. Total volume is 137 pages, 14 tables and 35 figures.

Content of the dissertation

In **Introduction** actuality of the study, its object and aim are defined. Further, the results of the study, its practical and scientific values are presented.

Chapter 1 "Use of artificial intelligence techniques while developing system specifications"

In this chapter an overview of the works in application of artificial intelligence for specification development is presented. The works are considered with respect to knowledge acquisition, intermediate and target representations, degree of automation and check of correctness in order to identify its general compound parts. Survey of the results in solving validation and verification problems in artificial intelligence is introduced. Generalised methods of static and dynamic analyses of knowledge bases are presented as well as the properties that are checked using these methods.

An approach for the creation of aggregate specifications is analysed and a technique for development of the specifications using knowledge bases is proposed. A schematic description of such a technique is presented in Figure 1.

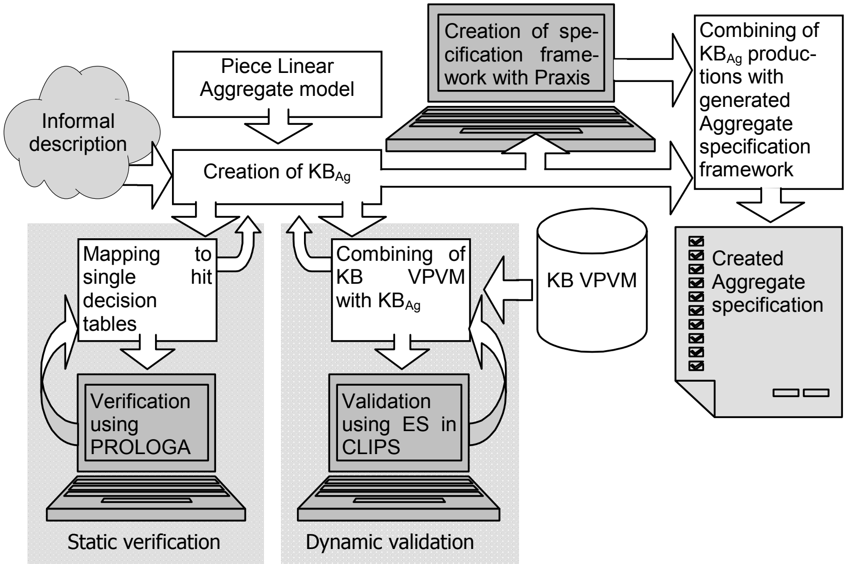

Figure 1. Scheme of the proposed approach for creation of Aggregate specifications

Knowledge base (KB) is created using the knowledge acquisition technique adapted for the creation of the specific KB. Created knowledge base is referred to as KB_{Ag}, and later mapped to aggregate specification. Knowledge about the problem domain is represented in the KB_{Ag} in the context of the Piece linear aggregate (PLA) model.

To perform static verification of KB_{Ag}, its system of production rules is mapped to a system of the single hit decision tables in PROLOGA. The PROLOGA is an interactive design tool for a computer-supported construction and manipulation of decision tables. It offers design techniques and additional features to enhance the construction and verification of decision tables. Mapping to PROLOGA decision tables (DTs) is specified with respect to PLA model and employs some of its concepts—aggregates, internal and external events, input and output signals, discrete state component coordinates, etc. Production rules are mapped to the DTs of certain groups thus enabling to fully exploit advantages of tabular representation to perform static verification.

Dynamic validation is performed using the expert system (ES) in CLIPS. CLIPS (C Language Integrated Production System) is a tool for the development and delivery of expert systems. The ES is constructed by combining the KB_{Ag} with the KB of validated properties and validation method (KB VPVM), where dynamic validation is implemented using the reachable states validation method. Further, using the validated and verified KB_{Ag}, aggregate specification framework is defined during the session with specification editor PRAXIS. The generated specification framework is supplemented with knowledge about behaviour of system extracted from the KB_{Ag} using defined mappings.

A distinctive feature of the proposed technique is the fact that validation and verification tasks are performed at the initial stage of the aggregate specification development. Validated knowledge is used both for creation of the specification framework and for supplementation to the framework.

In this chapter aims and tasks of the dissertation were formulated as well.

Chapter 2 “Development of the aggregate specification knowledge base”

In this chapter a technique for the creation of an aggregate specification knowledge base from an informal description is presented. Knowledge base uses non-pure production rule form, which permits to use: variables in condition and action parts, disjunctions and negations in condition parts of productions.

Informal description of the system is stored in a knowledge base. Concepts and relations are presented in KB_{Ag} in a form of predicates and production rules, which correspond to elements of PLA. The following objects are described in the aggregate specification knowledge base: input and output signals, operations, internal and external events, transition and output operators that describe state changes, interconnection scheme of aggregates. Below a fragment of KB_{Ag} structure is presented.

Input signal and its components are described by a predicate:

$$\texttt{InputSignal}\ (\ an_i, xn_i^j, x_i^{j,1}, \ldots, x_i^{j,l_j}\ ) \qquad (1)$$

This predicate describes an arrival of the *j*-th input signal with an identifier xn_i^j, which component values are $x_i^{j,1}, \ldots, x_i^{j,l_j}$, to the *i*-th aggregate an_i.

Production that describes an internal event (the end of operation) occurrence in an aggregate has the following form:

$$\begin{aligned}
&\texttt{If}\quad \texttt{State}(\ an_i, w_i^1, \ldots, w_i^{u-1}, w_i^u, w_i^{u+1}, ..., w_i^{C_i}, d_i^1, \ldots, d_i^{D_i}\ ) && (2)\\
&\qquad \texttt{and}\ \ w_i^u\ \ \texttt{= Active}\\
&\texttt{Then}\ \texttt{State}(\ an_i, w_i^1, \ldots, w_i^{u-1}, \texttt{Passive},\ \ , w_i^{u+1}, ..., w_i^{C_i}, d_i^1, \ldots, d_i^{D_i}\ )\\
&\qquad \texttt{and EndOfOperation}(\ an_i, wn_i^u\ )
\end{aligned}$$

where w_i^u - value of the *u*-th coordinate of continuous component of a state at the *i*-th aggregate, d_i^1 - value of the 1st coordinate of discrete component of a state at the *i*-th aggregate, wn_i^u - identifier of the *u*-th coordinate of continuous component of a state at the *i*-th aggregate.

Production that describes change of aggregate state after the occurrence of an internal event (end of operation) has the following form:

$$\begin{aligned}
&\texttt{If}\quad \texttt{EndOfOperation}(\ an_i, wn_i^u\ ) && (3)\\
&\qquad \texttt{and State}(\ an_i, w_i^1, \ldots, w_i^{C_i}, d_i^1, \ldots, d_i^{D_i}\ )\\
&\qquad \texttt{and Aux}(\ an_i, w_i^1, \ldots, w_i^{C_i}, d_i^1, \ldots, d_i^{D_i}\ )\\
&\texttt{Then}\ \texttt{State}(\ an_i, w_i^1*, \ldots, w_i^{C_i}*, d_i^1*, \ldots, d_i^{D_i}*\ )\\
&\qquad \texttt{OutputSignal}(\ an_i, yn_i^k, y_i^{k,1}, \ldots, y_i^{k,r_k}\ )
\end{aligned}$$

According to this production if the end of wn_i^u operation occurs and additional logical conditions on aggregate state coordinates, which are expressed in auxiliary predicate `Aux`, are satisfied then *i*-th aggregate state coordinates acquire new values that are marked with an asterisk and output signal is sent out.

An example of the creation of the aggregate specification knowledge base of a queuing system with priorities is presented in this chapter.

In this chapter, a relation between the proposed technique for development of the aggregate specification knowledge base and similar works is presented in a light of knowledge representation means, aims of the developed knowledge base and sources of knowledge, which are used for creation of knowledge bases.

Chapter 3 “Analysis techniques of static and dynamic properties of aggregate specification knowledge bases”

In the sub-chapter “*Static verification technique for aggregate specification knowledge bases*” technique that checks static properties of aggregate specification knowledge base is presented. It is based on single hit decision table representation formalism.

A decision table consists of four parts. The *condition subjects* are the criteria that are relevant to the decision making process. They represent the items about which information is needed to take the right decision. Condition subjects are found in the upper-left part of the table. The *condition states* are logical expressions determining the relevant sets of values for a given condition. Condition states are found in the upper-right part of the table. The *action subjects* describe the possible outputs of the decision making process. They are found in the lower-left part of the table. The *action values* are the possible values a given action can take. They are found in the lower-right part of the table. Every table column indicates which actions should (or should not) be executed for a specific combination of condition states.

In a single-hit table, each possible combination of condition states can be found in one and only one column. This exclusivity criterion is a key factor in verification, since it prevents most kinds of redundancy and ambivalence. Such representation is used in the PROLOGA system where static verification is performed.

The static verification technique for aggregate specification knowledge bases consists of two stages:

- a procedure to obtain a system of single hit decision tables from a system of productions of aggregate specification knowledge base;
- verification of anomalies (counterexample of general property) that are specific to KB_{Ag} using single hit decision tables.

A procedure for obtaining single hit decision tables from KB_{Ag} productions is given in the sub-chapter. The main parts of the procedure are the following:

1. A set of productions, which describe analysed system functioning in a case of a certain event, is mapped to a corresponding event decision table.
2. Condition subjects of the decision table are represented as:
    - predicates in antecedent part of a production rule;
    - state coordinates, which values are checked in auxiliary conditions of the production rule;

3. Condition states of the decision table are described for:
   - predicates - using Boolean values;
   - state coordinates - using conditions of the production rule;
4. Action subjects of the decision table are represented as predicates of consequent part of the production rule. While representing new values of state coordinates, the following is specified:
   - `not` (w_i^j `= Active`), if production consequent includes w_i^j `= Passive`;
   - `not` ($d_i^j{*} = d_i^j + Const$), if production consequent includes $d_i^j{*} = d_i^j - Const$ and productions describing the same event includes a pair $d_i^j{*} = d_i^j + Const$ and $d_i^j{*} = d_i^j - Const$.
5. Action values of the decision table are defined by predicates in production rule consequent.

In this procedure, use of separate tables for different events allows to perform static verification not only after the creation of the whole aggregate specification knowledge base of an analysed system, but also its fragment, which characterises system behaviour in a case of an event; the use of negation allows to check for the ambivalence in the aggregate specification knowledge base represented in a form of decision tables.

The procedure is illustrated by an example of the queuing system with priorities. Here only illustrative decision table is presented (see Table 1). Using the procedure presented, productions of type (3) are mapped to the following decision table (it is considered, 2 continuous and 2 discrete state coordinates describe aggregate *an*$_i$; ${}^{v1}d_i^1,\ldots,{}^{vm}d_i^1$ and ${}^{v1}d_i^2,\ldots,{}^{vk}d_i^2$ are all possible values for corresponding discrete coordinates; $f_{w_i^1}, f_{w_i^2}$ and $f_{d_i^1}, f_{d_i^2}$ are functions that define new values for corresponding state coordinates; action state as_j^t, $as_j^t \in \{\times, -, \cdot\}$ may acquire one of the values "execute", "do not execute", "undefined" depending on the represented production). A production rule may be represented by one or more DT columns.

Static verification is based on comparison of different parts of single hit decision table (one or several tables). During verification, decision tables are checked for anomalies that may be present in a single table or in inter-related tables. In the sub-chapter, a sub-set of general anomalies that are specific to aggregate specification knowledge base has been selected. Anomalies, which are checked during static verification of KB_{Ag}, are presented in the sub-chapter and summarised in Figure 2.

Table 1. Illustration of the procedure for mapping of KB_{Ag} productions to decision tables

1. EndOfOperation(an_i, wn_i^1)	True									False
2. State($an_i, w_i^1, w_i^2, d_i^1, d_i^2$)	True								False	-
3. w_i^2	Active							Passive	-	-
4. d_i^1	$^{v_1}d_i^1$			...	$^{v_m}d_i^1$			...	-	-
5. d_i^2	$^{v_1}d_i^2$	...	$^{v_k}d_i^2$	...	$^{v_1}d_i^2$	...	$^{v_k}d_i^2$	...	-	-
1. $w_i^{1*} = f_{w_i^1}$	as_1^1	...	...	...	...	...	...	...	.	.
2. $w_i^{2*} = f_{w_i^2}$	as_2^1	...	...	...	...	...	...	...	.	.
3. $d_i^{1*} = f_{d_i^1}$	as_3^1	...	...	...	...	...	...	...	.	.
4. $d_i^{2*} = f_{d_i^2}$	as_4^1	...	...	...	...	...	...	...	.	.
5. State($an_i, w_i^1, w_i^{2*}, d_i^{1*}, d_i^{2*}$)	as_5^1	...	...	...	...	...	...	...	.	.
	1	...	k	...	...	...	...	...	...	

- redundancy
  - subsumed column pair
    - duplicate column pair
  - unsatisfiable column
    - unsatisfiable condition state
  - irrelevant condition row
  - unreferenced action row
- ambivalence
  - ambivalent column pair
  - ambivalent action rows
- deficiency
  - missing column
    - unused condition value
  - unknown action state

Figure 2. Anomalies that are checked during the static verification of KB_{Ag}

In this way, KB_{Ag} expressed in single hit decision tables has

- the special case of *redundancy* anomaly:
  - *subsumed column pair* - if several KB_{Ag} productions correspond to the same table column;
  - *duplicate column pair* - if the same KB_{Ag} productions correspond to the same table column;

   - *unsatisfiable column* - if column condition state is out of range of allowed values;
   - *irrelevant condition row* - if the functions represented by the decision table do not depend on the corresponding condition, i.e. all condition states in the row are marked "irrelevant";
   - *unreferenced action row* - if certain action entry is not used in conditional part of another table;
- the special case of *ambivalence* anomaly:
   - *ambivalent column pair* - if two table columns have contradictory action states or these states violate semantic constraint(s);
   - *ambivalent action rows* - if two table action rows with identical subjects have different indications of action.
- the special case of *deficiency* anomaly:
   - *missing column* - if possible input of a decision table does not covered by corresponding condition state;
   - *unknown action state* - if a set of condition states corresponding to internal/external event is covered only by actions with state indicators "undefined".

All KB_{Ag} decision table anomalies presented in the sub-chapter are illustrated by an example of the queuing system with priority. Identified anomalies in decision tables have corresponding anomalies in productions of KB_{Ag} (also presented in the sub-chapter).

In the sub-chapter "*Dynamic validation technique for aggregate specification knowledge bases*" technique for checking dynamic properties of aggregate specification knowledge base is presented. Dynamic properties are checked using a dynamic validation expert system (DVES). It is constructed by joining:

- verified aggregate specification knowledge base with
- knowledge base of validated properties and validation method (KB VPVM).

A combining procedure, as well as a proposed content of KB VPVM, is presented in the sub-chapter. KB VPVM, also as KB_{Ag}, has a production rule representation and contains descriptions of the following validated general dynamic properties:

- absence of static deadlocks;
- reachability;
- boundedness of state coordinates;
- absence of over specification.

Reachable states validation method and forward chaining mechanism are used in DVES. Reachable states validation method provides a schema for the generation of all possible model execution paths while analysing impacts on global states of the model during its functioning. Breadth first strategy is used to generate global states of an analysed system. Therefore global states generated from the same ancestor are analysed first. Suggested productions that describe reachable states validation method are presented in the sub-chapter.

General scheme of dynamic validation of aggregate specification knowledge base is presented in the sub-chapter (see Figure 3).

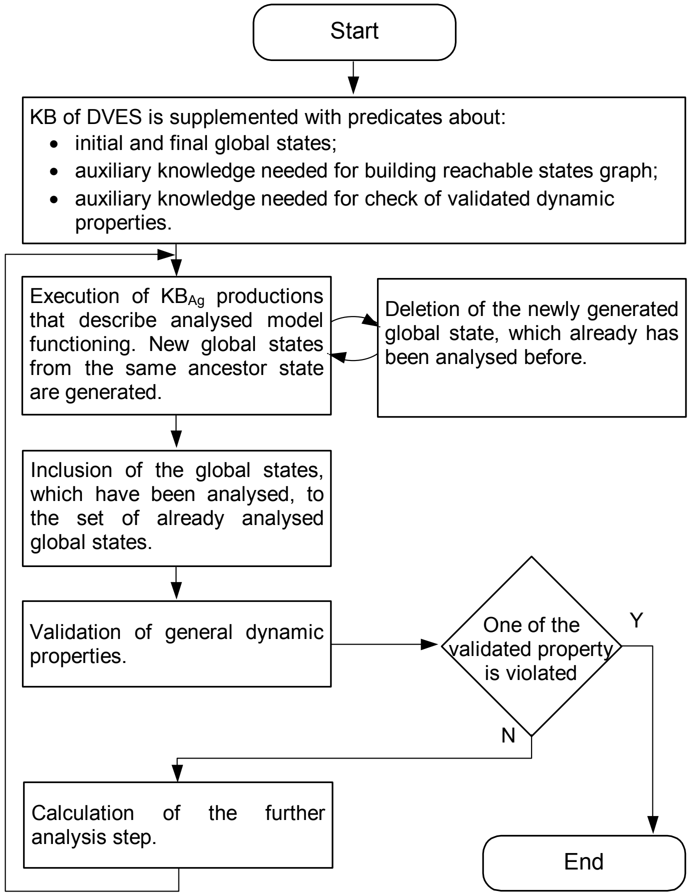

Figure 3. General scheme of dynamic validation of aggregate specification knowledge base

The experiments illustrating application of dynamic validation technique are presented in the sub-chapter. The experiments of single channel queuing systems have also demonstrated that analysis of static properties additionally allows checking redundancy (and its special cases: subsumed and duplicated rules, an irrelevant condition) and ambivalent rules as well as deficiency.

In this chapter, a relation of the proposed techniques for analysis of static and dynamic properties of aggregate specification knowledge base with other similar works is presented in a view of representation techniques and validation methods. Particular attention is paid to the comparison of the technique for KB_{Ag} analysis proposed in the dissertation and the analysis of aggregate specification knowledge bases based on the first order predicates and backward inference mechanism. Various aspects of techniques were compared: knowledge representation, target source of knowledge, inference mechanism, types of general properties validated and execution times of model implementations of dynamic validation.

Chapter 4 "Join of productions of aggregate specification knowledge base with the structure of aggregate specification"

In the chapter predicates of aggregate specification knowledge base, used to define the framework of the specification while working with PRAXIS system, are presented. A general view of the generated specification framework is presented in Figure 4. The procedure of mapping of predicates and productions of validated and verified aggregate specification knowledge base into the framework of generated aggregate specification is presented later in the chapter.

The framework consists of a set of modules M, $\mathrm{M} = \{ M_1, M_2, ..., M_n \}$. Every module M_i consists of constructs that describe its initial state (`Initialize` construction) and transition between states (`Trans` construct): $M_i = \{I_i, T_i\}, i = \overline{1,n}$ where I_i and T_i correspond to `Initialize` and `Trans` constructs respectively.

Predicates and productions of validated KB_{Ag} with the generated aggregate specification framework are combined in several phases:

1. Module $M_i, i = \overline{1,n}$ is selected in the generated framework of aggregate specification. Let this module has symbolic identifier an_i.
2. Construct $t_{i,j}$ of an_i module is analysed $t_{i,j} \in T_i, j = \overline{1,(C_i + N_i)}$, where C_i - number of coordinates of continuous state component, N_i - number of input signals at the i-th aggregate.

```
Specification spec;
  ...
  Module an_i;
    ...
    Initialize
      begin
      end;
    Trans
      when iip_i^j.iid_u^t
        begin
        end;
      when eop.wn_i^j
        begin
        end;
      ...
  end;
  Module an_j;
    ...
  end;
  ...
end.
```

Figure 4. A general form of the generated specification framework

If construct $t_{i,j}$ describes an internal event (the end of operation), i.e. corresponds to `when eop.` wn_i^u `begin end` form, then:

2.1 Production `r`, which conditional part includes predicate `EndOfOperation(` an_i, wn_i^j `)`, is selected from KB_{Ag} rule base. Values of arguments in the predicate have to be the same as ones that are identified in the considered `when` construct. The selected rule is specified in the considered construct `when` in a form of conditional operator `If ... Then`, which condition/action parts are defined later.

2.2 Auxiliary conditions for coordinates of the global state, that are defined in predicate `Aux` of the selected rule `r`, are specified in condition part of the operator `If ... Then` in the construct `when`.

2.3 Predicates in the action part of the analysed rule are considered. Values of coordinates of the new state that are defined in the predicate `State` are specified in assignment operators in the action part of operator `If …Then`. Arguments $oip_i^j, iid_u^t, y_i^{j,1}, \ldots, y_i^{j,r_j}$ of the predicate `OutputSignal` are specified in the action part of operator `If … Then` in the following way: `Output` $oip_i^j.iid_u^t\left(y_i^{j,1},\ldots,y_i^{j,r_j}\right)$.

2.4 Items 2.1 - 2.3 are repeated until KB_{Ag} rule base contains productions that satisfy requirements of item 2.1.

If construct $t_{i,j}$ describes an external event (arrival of input signal to interaction point iip_i^j of aggregate an_i through channel interaction point iid_u^t), i.e. corresponds to `when` $iip_i^j.iid_u^t$ `begin end` form, then:

2.5 Production `r`, which conditional part includes predicate `External_Event(` $an_i, sn_i^j, iip_i^j, ln_u, iid_u^t$ `)`, is selected from KB_{Ag} rule base. Values of iip_i^j, iid_u^t arguments in this predicate are the same as identified in the considered `when` construction. The selected rule is specified in the considered construct `when` in a form of a conditional operator `If ... Then`, which condition/action parts are defined later.

2.6 the same as item 2.2.

2.7 the same as item 2.3.

2.8 Items 2.5 - 2.7 are repeated until KB_{Ag} rule base contains productions that satisfy requirements of item 2.5.

3. Construction `Initialize` of an_i module is analysed. Production `r`, which conditional part contains predicate `Start(`an_i`)`, is selected from KB_{Ag} rule base. Values of coordinates of the initial state, which are defined in the predicate `State` in the action part of the selected rule, are specified in assignment operators.

An example of the proposed technique application is presented in this chapter. An aggregate specification of queuing system with priorities is formed by combining generated framework of the specification with productions and predicates of the specification knowledge base, which was developed, validated and verified earlier.

In the chapter it is concluded that combining of generated specification framework with productions and predicates of the specification knowledge base allows constructing aggregate specification which algorithmic part was validated and verified using the suggested techniques.

Chapter 5 "Examples of creation and analysis of aggregate specifications"

Illustration of the presented techniques is provided in the chapter. Aggregate specifications of alternating bit protocol, network of queuing systems and Internet cache protocol (industrial size application) are developed and analysed.

Aggregate specification knowledge bases were derived from informal descriptions. The proposed techniques were used to check static and dynamic properties such as absence of redundancy, ambivalence and deficiency and their special cases (static properties) and absence of static deadlocks, reachability, boundedness of state coordinates, absence of over specification (dynamic properties). Redundancy and ambivalence were detected in KB_{Ag} of the Internet cache protocol. These occurrences were fixed. Any other violations of analysed properties were not detected.

Validated and verified aggregate specification knowledge bases were used while defining frameworks of the corresponding specifications in PRAXIS system as well as while combining its predicates and productions with corresponding generated specification frameworks.

Results of executed experiments show that the application of the proposed techniques allows to represent analysed systems using their description concepts and to check additional properties — absence of redundancy, ambivalence and deficiency. These static properties were not checked before.

Conclusions

1. The technique proposed in the dissertation allows to validate general static and dynamic properties of aggregate specifications during development process and to develop these specifications using knowledge bases. Advantages of the technique are evaluated using the following criteria:
    - *preliminary validation* - possibility to perform validation and verification experiments at the initial specification development stages when a knowledge base of the specification is being developed;
    - *check for additional properties* - properties, that have not been examined during validation of aggregate specifications, are analysed during static verification.
2. Technique based on single hit decision tables for static verification of the aggregate specification knowledge bases was developed. The technique allows examining absence of redundancy and ambivalence and deficiency. These properties were not checked before.

3. Using the developed technique of static verification of aggregate specification knowledge bases, general static properties such as absence of redundancy, ambivalence and deficiency can be examined not only after creation of a whole specification knowledge base of an analysed system but having represented only its fragment defining behaviour of the system on occurrence of an event.
4. Created technique for validation of general dynamic properties, such as reachability, boundedness of state coordinates, absence of over specification and static deadlocks of aggregate specification knowledge base may be made by means of production rule representation and forward chaining.

List of scientific publications on the theme of the dissertation

Publications inscribed in the main list of Institute of Scientific Information (*ISI*)

1. H. Pranevicius, G. Budnikas. Creation of Estelle/Ag specifications using knowledge bases. *Informatica*, Vol. 14, No. 1, 2003: pp. 63-74. ISSN 0868-4952.

Publications included in the list approved by the Department of Science and Study

1. H. Pranevicius, G. Budnikas. Static verification of aggregate specifications. *Information technology and control*, No. 4(29), Kaunas, „Technologija", 2003: pp. 39-46. ISSN 1392-124X.
2. H. Pranevicius, G. Budnikas. The Use of Knowledge Representation Models for ESTELLE/Ag Specification Creation. *Information technology and control*, No. 2(3), Kaunas, „Technologija", 1996: pp. 29 - 33. ISSN 1392-124X.

Publications in other reviewed international and foreign editions

1. H. Pranevicius, G. Budnikas. Use of Knowledge Engineering Techniques for Creation and Analysis of Aggregate Specifications. *Sixth International Baltic Conference on Databases & Information Systems 2004*. Scientific papers University of Latvia, Volume 672, Databases and Information Systems 2004: pp. 309-320. ISSN 1407-2157.
2. G. Budnikas. Application of knowledge-based techniques for creation of ESTELLE/AG specifications. In proc.of the International Conference Modelling and Simulation of Business Systems, Eds. H. Pranevicius, E. Zavadskas, B. Rapp. Kaunas, „Technologija", 2003: pp. 172-176. (*ISI Proceedings*) ISBN 9955-09-420-6.

3. H. Pranevicius, G. Budnikas. Use of temporal logic for reachability analysis of aggregate specifications. // Simulation Applications in the Baltic Area. Research Report Series, Report no. 5, 2002: pp. 151-157. ISSN 1404-2738.

Publications in proceedings of Lithuanian conferences

1. G. Budnikas. Žinių modelių verifikavimas panaudojant tarpusavyje susietas sprendimo lenteles. Informacinės technologijos 2000, Kaunas, „Technologija", 2000: pp. 190-197. ISBN 9986-13-826-4.

Information about the author of the dissertation

1990 - 1994: Studies at Kaunas University of Technology, Informatics faculty - Bachelor of Sciences in Informatics.

1994 - 1996: Studies at Kaunas University of Technology, Informatics faculty - Master of Sciences in Informatics (*diploma with distinction*).

1999 - 2003: PhD studies in Physical Sciences (Informatics 09P) at Kaunas University of Technology, Faculty of Informatics, Business Informatics Department.

Reziumė

Tyrimo objektas – agregatinės specifikacijos sudaromos iš neformalių paskirstytųjų sistemų aprašų.

Temos aktualumas. Paskirstytosios sistemos tampa vis sudėtingesnės kaip ir architektūros, kuriomis jos remiasi. Programų sistemos vis dažniau funkcionuoja aplinkoje, kuri keičiasi atsižvelgiant į vartotojų poreikius. Vienas šių problemų sprendimo būdų yra formaliųjų metodų panaudojimas sistemų projektavimo etape, kai kuriami ir analizuojami šių sistemų modeliai. Viena formaliųjų metodų klasių – laiko automatai. Agregatinis specifikavimo metodas priklauso šiai klasei. Įrankių rinkinys PRANAS taiko formalųjį agregatinį specifikavimo metodą programinės įrangos sistemų kūrimo etape – sudarant ir analizuojant agregatinę specifikaciją. Sistema PRANAS sukurta Kauno technologijos universiteto Verslo informatikos katedroje. Šio rinkinio komponentas – specifikacijų redaktorius, – automatiškai generuoja agregatinės specifikacijos struktūrą. Ji susideda iš sistemos modulių aprašymo bei būsenų pasikeitimo sąlygų. Sugeneruota specifikacijos struktūra euristiniu būdu papildoma nepatikrintu algoritminės dalies aprašymu apie sistemos funkcionavimą keičiantis būsenai.

Sistemos sudarymas paprastai prasideda nuo analizuojamojo objekto aprašo, kuriam kurti naudojamos naudojant tik simbolinės reikšmės. Žiniomis grįstoms

sistemoms ir priemonėms taikomas šis būdas. Sistemos specifikacija gali būti pateikta žinių apie nagrinėjamą sritį forma, nenaudojant tikslių skaitmeninių verčių. Pradinis aprašas gali būti ne iki galo sudarytas, prieštaringas ir netikslus. Taigi būtinos metodikos bei priemonės nesuderinamumams aptikti pradiniame sistemos kūrimo etape, kai kuriama formali specifikacija, nagrinėjamuoju atveju – agregatinė specifikacija.

Tyrimo tikslas – sudaryti agregatinių specifikacijų kūrimo ir statinių bei dinaminių savybių tikrinimo naudojant žinių bazes metodiką. Tyrimo uždaviniai:

- Agregatinės specifikacijos žinių bazės agregatinėms specifikacijoms kurti, sudarymas iš neformalaus aprašymo.
- Agregatinės specifikacijos žinių bazės, iš kurios bus sudaroma agregatinė specifikacija, statinis verifikavimas ir dinaminis validavimas.
- Agregatinės specifikacijos sudarymas naudojant validuotą ir verifikuotą specifikacijos žinių bazę.
- Sukurtos metodikos patvirtinimas eksperimentais, sudarant šių sistemų modelių agregatines specifikacijas: aptarnavimo sistemos su prioritetais, alternuojančiojo bito protokolo, aptarnavimo sistemų tinklo, interneto spartinančiosios atminties protokolo.

Tyrimo metodai bei programinės priemonės:

- Agregatinis specifikavimo metodas.
- Nepasikartojančių situacijų sprendimo lentelių verifikavimo bei pasiekiamų būsenų validavimo metodas.
- Agregatinių specifikacijų redaktorius PRAXIS; žinių bazių statinio verifikavimo sistema PROLOGA; ekspertinių sistemų kūrimo ir analizės įrankis CLIPS žinių bazių dinaminiam validavimui vykdyti; agregatinių specifikacijų validavimo sistema PRANAS.

Ginamieji disertacijos teiginiai. Panaudojant žinių inžinerijos priemones, sukurta metodika, kuri leidžia sudaryti agregatinių specifikacijų algoritminę dalį iš anksto patikrinus jos statines ir dinamines savybes:

1. Agregatinės specifikacijos žinių bazės, kuri pritaikyta agregatinėms specifikacijoms sudaryti, struktūra naudojant produkcines taisykles. Šioje bazėje aprašomi agregatinį modelį apibrėžiantys objektai: įvesties ir išvesties signalai, operacijos, vidiniai bei išoriniai įvykiai, perėjimo bei išvesties operatoriai, agregatų sujungimo schema.
2. Agregatinės specifikacijos žinių bazės statinio verifikavimo metodika naudojant sprendimo lenteles. Ji leidžia tikrinti šias savybes: pertekliškumo, prieštaringumo bei nepakankamumo nebuvimą. Remiantis šia metodika pažymėtos statinės savybės taip pat gali būti tikrinamos ne vien tik sudarius visą

nagrinėjamos sistemos agregatinės specifikacijos žinių bazę, bet pavaizdavus tik jos fragmentą, kuris apibūdintų sistemos funkcionavimą įvykiui įvykus.

3. Agregatinės specifikacijos žinių bazės bendrųjų dinaminių savybių, – pasiekiamumo, būsenos koordinačių apribojimų tenkinimo, statinių aklaviečių ir pertekliškumo nebuvimo, – validavimo metodika taikant produkcinių taisyklių vaizdavimo bei tiesioginio išvedimo būdus.
4. Sukurtos, validuotos bei verifikuotos žinių bazės, kurios panaudotos aptarnavimo sistemos su prioritetais, alternuojančiojo bito protokolo, aptarnavimo sistemų tinklo, interneto spartinančiosios atminties protokolo agregatinėms specifikacijoms sudaryti taikant pasiūlytą agregatinių specifikacijų sudarymo ir analizės metodiką.

Darbo praktinė reikšmė. Žinių bazės, kurios naudojamos, sudarant agregatines specifikacijas, leidžia aprašyti specifikacijos algoritminę dalį analizuojamosios sistemos aprašo konceptais bei ryšiais. Pasiūlytoji metodika leidžia tikrinti nagrinėjamosios sistemos modelio statines bei dinamines savybes specifikacijos sudarymo etape.

Darbo mokslinis naujumas. Pasiūlytoji metodika, naudojanti žinių inžinerijos priemones, leidžia aprašyti kuriamos agregatinės specifikacijos algoritminę dalį analizuojamosios sistemos aprašo terminais bei vykdyti statinių savybių patikrą specifikacijos sudarymo etape.

Publikacijos. Darbo tema paskelbtos 7 mokslinės publikacijos:

- Leidiniuose, įrašytuose į Mokslinės informacijos instituto pagrindinį sąrašą (*ISI*) 1;
- Lietuvos leidiniuose, įrašytuose į Mokslo ir studijų departamento patvirtintą sąrašą 2;
- Kituose recenzuojamuose tarptautiniuose ir užsienio leidiniuose 3;
- Lietuvos konferencijų pranešimų medžiagose 1.

Išvados

1. Pasiūlytoji metodika leidžia tikrinti kuriamų agregatinių specifikacijų bendrąsias statines ir dinamines savybes bei sudaryti šias specifikacijas naudojant žinių bazes. Metodikos pranašumai vertinami šiais kriterijais:
   - *išankstinis validavimas* – galimybė vykdyti validavimo bei verifikavimo eksperimentus pradiniuose specifikacijos sudarymo etapuose, kai sudaroma šios specifikacijos žinių bazė;
   - *papildomų savybių patikra* – statinio verifikavimo metu tikrinamos savybės, kurios iki šiol nebuvo analizuojamos tikrinant agregatines specifikacijas.

2. Agregatinės specifikacijos žinių bazės statinio verifikavimo metodika, sudaryta naudojant nepasikartojančių situacijų sprendimo lenteles, leidžia tikrinti bendrąsias statines savybes, – pertekliškumo, prieštaringumo bei nepakankamumo nebuvimą, – kurios nebuvo tikrinamos iki šiol.
3. Remiantis agregatinės specifikacijos žinių bazės statinio verifikavimo metodika pertekliškumo, prieštaringumo bei nepakankamumo nebuvimo bendrosios statinės savybės gali būti tikrinamos ne vien tik sudarius visos nagrinėjamos sistemos agregatinės specifikacijos žinių bazę, bet ir pavaizdavus tik jos fragmentą, kuris apibūdintų sistemos funkcionavimą įvykiui įvykus.
4. Agregatinės specifikacijos žinių bazės bendrųjų dinaminių savybių, – pasiekiamumo, būsenos koordinačių apribojimų tenkinimo, statinių aklaviečių ir pertekliškumo nebuvimo, – validavimo metodika gali būti sudaryta naudojant produkcinių taisyklių vaizdavimo bei tiesioginio išvedimo būdus.

UDK 004.414.28 + 004.41 + 004.048 + 004.415.5] (043)

SL 344. 2004-11-15, 1 leidyb. apsk. l. Užsakymas 423, Tiražas 70 egz.
Išleido leidykla „Technologija", K. Donelaičio g. 73, 44029 Kaunas
Spausdino leidyklos „Technologija" spaustuvė, Studentų g. 54, 51424 Kaunas