

# Creation of ESTELLE/Ag Specifications Using Knowledge Bases

Henrikas PRANEVICIUS, Germanas BUDNIKAS

*Business Informatics Department, Kaunas University of Technology  
Studentų 56–301, LT-3031 Kaunas, Lithuania  
e-mail: hepran@if.ktu.lt*

Received: January 2003

**Abstract.** The paper presents a technique that defines creation of ESTELLE/Ag specifications using knowledge bases (KB). Application KB is created using the knowledge acquisition technique joined with a piece-linear aggregate model. The production rules of the application KB are transformed to decision tables, and the static properties of the KB are checked in PROLOGA system. Further, the application KB is combined with the defined KB of validated properties and validation method, and application KB dynamic properties are checked in the expert system in CLIPS. A validated application KB is used defining a framework of ESTELLE/Ag specification using PRAXIS editor and supplementing PRAXIS generated framework with the application functional description. The technique is illustrated with an example of a single channel queuing system.

**Key words:** ESTELLE/Ag specification, knowledge base, expert system, validation of static and dynamic properties, CLIPS, PROLOGA.

## 1. Introduction

Formal methods and specification languages are widely used for design of distributed systems. The most popular formal specification languages being used for description of distributed systems are SDL, LOTOS, ESTELLE (Facchi *et al.*, 1996; Spirakis *et al.*, 1996). In (Pranevičius *et al.*, 1994), a motivation for specification in ESTELLE/Ag language is presented. The main advantage of this language is its ability to perform both simulation and validation tasks on the basis of single specification.

ESTELLE/Ag specifications are based on a piece-linear aggregate (PLA) model (Pranevičius *et al.*, 1994). A construction of specifications in this language is performed in two phases. Specification editor PRAXIS helps to define the specification framework that describes an interaction of specified system aggregates, their states and conditions of state change. Next, the framework is heuristically supplemented with the knowledge from a conceptual model about behaviour of an analysed system.

The works where knowledge-based systems (KBS) are used for creation of (formal) specifications include: acquisition of requirements and incremental evolution of specifications (Johnson *et al.*, 1991); definition of requirements for transition from an informal problem description to formal specification in VDM language (d'Alameida *et al.*,

1992); generation of ESTELLE/Ag specification structure (Pranevičius *et al.*, 1994); use of knowledge-based meta-model for integrating characteristics of software development models (Mi and Scacchi, 1995); formulation of requirement specifications in domain concepts (Fuchs and Schwitter, 1996); use of formalisation step for problem knowledge description (Sellini *et al.*, 1997); database design by creating its specification (Noah and Williams, 2000) and definition of system requirements and analysis (Vanthienen, 2000). General parts of such techniques/tools are:

- the knowledge acquisition that may be realised using text translators (e.g. (Johnson *et al.*, 1991; Fuchs and Schwitter, 1996; Noah and Williams, 2000)), or assisting by Graphic User Interface (e.g. (d'Alameida *et al.*, 1992; Pranevičius *et al.*, 1994; Vanthienen, 2000));
- transformation of intermediate discourse representation structures (like Modified ER diagrams (d'Alameida *et al.*, 1992); restricted English (Fuchs and Schwitter, 1996), etc.) to a well-defined model (e.g. object-oriented model (Mi and Scacchi, 1995; Noah and Williams, 2000); first order predicates (Fuchs and Schwitter, 1996)) or target specification (e.g. VDM (d'Alameida *et al.*, 1992); ESTELLE/Ag (Pranevičius *et al.*, 1994)).

A key feature of these techniques is the analysis of correctness that is performed both during the construction of the initial knowledge description (e.g. (Noah and Williams, 2000; Sellini *et al.*, 1997)) and while a target model or KB has been created (e.g. (Johnson *et al.*, 1991; Mi and Scacchi, 1995; Fuchs and Schwitter, 1996) and (Vanthienen, 2000)).

This paper presents an approach that defines a transition from an informal conceptual description of an application to its ESTELLE/Ag specification using knowledge bases. Creation of an application KB and its addition to the generated ESTELLE/Ag specification framework is defined. In our approach, the application KB is validated<sup>1</sup> for static and dynamic properties.

The static properties are characteristics of a KBS that can be evaluated without its execution. Such an evaluation is often referred to as static or structural verification. During static verification, a KB is checked for anomalies (Meseguer and Preece, 1996). Preece and Shinghal (1994) present a classification of the anomalies that may be present in rule-based systems. The dynamic properties are those characteristics of a rule-based system that can be evaluated only by examining how the system operates at a run time. The most common techniques of validation and verification that have been developed for use on KBS are identified in (Preece, 2001). A detailed review of specific methods and supporting tools can be found in (Gupta, 1990; Preece *et al.*, 1992; Miller *et al.*, 1993; Preece and Shinghal, 1994).

KB and ESTELLE/Ag specification are used for description of the same problem. Many authors (e.g. Schreiber *et al.*, 1993; Bruynooghe *et al.*, 1999) believe that the declarative style of description that is used in KBs is more understandable and acceptable than the procedural style (the latter is used in ESTELLE/Ag specifications). This is

---

<sup>1</sup>In the paper, for the sake of brevity, sometimes we refer to validation having in mind both validation and verification because "validation subsumes verification" (Preece *et al.*, 1996).

because a problem is described at the knowledge level “at which the knowledge engineer specifies expertise during knowledge acquisition” (Velde and Aamodt, 1994). Therefore, the knowledge description is presented not using strict mathematic notation but concepts of an application domain that is natural. Due to this reason, we suppose that the creation of ESTELLE/Ag specifications using KBs is more attractive in that sense.

Until now, knowledge-based techniques were not used for the creation of ESTELLE/Ag specifications except PRAXIS system. However, this system generates structure of the specification only and does not define the behaviour of an application. Our approach does it by using KBs.

Our technique is similar to the one proposed by Johnson *et al.* (1991), Fuchs and Schwitter (1996) in such a way that they also offer transformation of problem domain description to representation structures and then to an executable language. However, our technique checks general properties during validation and verification while Fuchs and Schwitter (1996) technique checks specific invariant properties. Our approach is similar to that of (Specht, 1998) since they both offer the use of declarative languages for a description of the user needs. We use aggregate model concepts for representation of the object structure, while Specht (1998) use “object as theory” model. In our approach, in contrast to the compared one, we emphasise validation of the created declarative description. A transformation to target representation is used in both approaches. Sen *et al.* (2002) present a technique for building the declarative descriptions in a unified way. We solve the same problem. Our and (Mi and Scacchi, 1995) approaches are comparable in a viewpoint they both offer to check static and dynamic properties. General structures of the facts and rules for KBs of application and validation are defined in both approaches too. Sellini *et al.* (1997) and we use intra- and inter- validation for analysis of the acquired knowledge. They also use an intermediate formalised description (application KB in our case) for construction of a knowledge model (specification in our case). While performing static verification of an application KB, we use results of Vanthienen *et al.* (1997), Vanthienen (2000), whereas when analysing the dynamic properties we use the reachable state method that is similar to the functional validation method suggested by Preece *et al.* (1996) in a view of analysis of execution paths.

An applicability of our technique is defined by the applicability of the PLA method and ESTELLE/Ag specifications. They are mostly used for formal specification and analysis of telecommunication protocols. Our technique was successfully applied for the creation of ESTELLE/Ag specifications of the single channel queuing system, alternating bit protocol and a network of queuing systems. The scalability of the proposed technique is limited by software tools that are used in creating the specification. The limitation requirements for these tools are defined in (Giarratano, 1993) and (Vanthienen, 2000).

The paper is structured as follows. The main stages of the proposed scheme of ESTELLE/Ag specification creation are presented in Section 2. Section 3 describes construction of the application KB and analysis of its static and dynamic properties. Section 4 describes procedures for generation of specification structure with PRAXIS and addition of functional description of the application from the KB to the generated structure. Conclusions sum up the proposed approach.

## 2. The Main Stages of the Proposed Scheme

The developed approach is depicted in Fig. 1. The application KB is created using the knowledge acquisition technique that was adapted for the creation of the specific KB (intended for mapping to ESTELLE/Ag specification). Knowledge about the problem domain is represented in the application KB in the context of the PLA model.

Production rules of the application KB are transformed to PROLOGA decision tables, and static verification is performed in this system. The PROLOGA system is an interactive design tool for computer-supported construction and manipulation of decision tables. The system offers design techniques and additional features to enhance the construction and validation of decision tables (Vanthienen, 2000). The verification in PROLOGA is implemented using the tabular verification method (Vanthienen *et al.*, 1997) that belongs to a group of static verification methods. Functional validation is performed using the expert system (ES) in CLIPS. CLIPS (C Language Integrated Production System) is a tool for productive development and delivery of expert systems (Giarratano, 1993). The ES is constructed by combining the application KB with the knowledge base of validated properties and validation method (KB VPVM). The functional validation is implemented using the reachable state validation method that can be classified as being a member of the group of formal proof methods. Further, using validated and verified application KB one defines ESTELLE/Ag specification framework during the session with specification editor PRAXIS. The generated framework is supplemented with knowledge about system behaviour extracted from the application KB using defined mappings.

A distinctive feature of our approach is the fact that validation and verification task is performed at the initial stage of ESTELLE/Ag specification creation. Validated knowledge is used both for creation of the specification framework and for supplementation to the framework. In addition, our approach defines the creation of ESTELLE/Ag specifications.

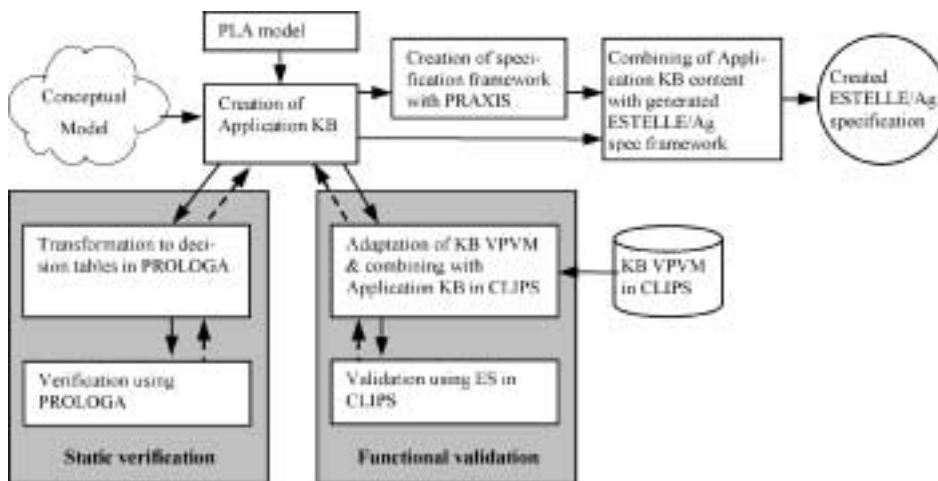


Fig. 1. Scheme of the proposed approach for creation of ESTELLE/Ag specifications.

### 3. Creation and Analysis of Application KB

The application KB represents problem domain knowledge in production rules formalism. This representation is chosen due to its similarity to ESTELLE/Ag specification language constructions describing conditions for state change: `when condition begin ... end`. Since most of the common verification and validation problems in rule-based systems can be solved using decision tables (Vanthienen and Wets, 1995) and the tabular verification method is computerised in PROLOGA system, in our approach static verification will be performed using this method. Thus, in order to perform the static verification of the KB, its production rules have to be transformed to PROLOGA decision tables. Moreover, as stated in (Vanthienen and Wets, 1995), a decision table is equivalent to a set of production rules, and their transformation to the tables can be performed without too much effort.

#### 3.1. Creation of Application KB

Because the application KB will be used for the creation of ESTELLE/Ag specifications, it has to contain knowledge about the PLA model. To acquire this knowledge, we applied the knowledge acquisition technique (Russel and Norvig, 1995) that was adapted for our needs. The following knowledge about the PLA model is used.

- Succeeding concepts: (a) Aggregates; (b) Input and output signals and their components, coordinates of discrete and continuous state components, operations with queues.
- Relations: (a) Interconnection scheme of aggregates; (b) Changes of state and signal outputs.

In our approach, the facts and production rules are used for description of the above-mentioned concepts and relations. Applying our approach for specification of an application one has to use the defined facts and production rules for problem representation. Thus, the application KB is created by filling in the knowledge about the application as facts and production rules of the defined form.

Further, we illustrate creation of the application KB with the example of the single channel queuing system (QS). For the description the CLIPS language syntax is used. This choice is because in that system the functional validation is performed.

```
(defrule r1 "arrival of request and QS is empty"
  (ArrivalOf "incoming request")
  (InputSignal "incoming request")
  (State ?request_processing ?query_size)
  (= ?query_size 0)
  (= ?request_processing off)
=>
  (bind ?request_processing on)
  (State ?request_processing ?query_size))

(defrule r2 "arrival of request and QS is busy"
  (ArrivalOf "incoming request")
  (InputSignal "incoming request")
  (State ?request_processing ?query_size)
  (or (> ?query_size 0)
      (= ?request_processing on))
=>
  (QueueOperation "request" put)
  (bind ?query_size (+ ?query_size 1))
  (State ?request_processing ?query_size))
```

### 3.2. Static Verification of the Application KB

In order to perform static verification of the application KB, its production rules are transformed to PROLOGA decision tables and verification is performed using this system. These topics are discussed further.

#### 3.2.1. Transformation of Application KB Production Rules to PROLOGA Decision Tables

A decision table consists of four parts: condition subjects, condition states, action subjects, and action values. These parts are formally described in (Vanthienen and Wets, 1995). The transformation is performed according to the following rules:

- Facts before ‘=>’ sign (left hand side– LHS) of a production rule are written as condition subjects of the decision table;
- Condition states for condition subjects are specified according to the conceptual model and the considered production rule;
- Facts on the right hand side (RHS) of a production rule are written as action subjects of the decision table;
- All production rules are represented as *definitely if* rules. That means, these rules cannot be overridden by other rules. This is checked by PROLOGA system.

The PROLOGA decision table depicted in Fig. 2 illustrates representation of production rule  $r_2$  of our example. Once transformation to PROLOGA decision tables has been performed, verification experiments are executed.

	TRUE	FALSE		
1. (ArrivalOf "incoming request")				
2. (InputSignal "incoming request")				
3. (State ?request_processing ?query_size)	TRUE	FALSE	-	-
4. ?query_size	=0	>0	-	-
5. ?request_processing	on	off	-	-
1. (QueueOperation "request" put)	x	x		
2. (bind ?query_size (+ ?query_size 1))	x	x		
3. (State ?request_processing ?query_size)	x	x		
	1	2	3	4

Conditions:

- c1-1 (ArrivalOf "incoming request")
  - a TRUE
  - b FALSE
- c1-2 (InputSignal "incoming request")
  - a TRUE
  - b FALSE
- c1-3 (State ?request\_processing ?query\_size)
  - a TRUE
  - b FALSE
- c1-4 ?query\_size
  - a =0
  - b >0
- c1-5 ?request\_processing
  - a on
  - b off

Actions:

- 1. (QueueOperation "request" put)
- 2. (bind ?query\_size (+ ?query\_size 1))
- 3. (State ?request\_processing ?query\_size)

Rules:

1. 1 and 2 and 3 definitely if (1a and 2a and 3a and (4b or 5a))

Fig. 2. Illustration of representation of production rule  $r_2$  as PROLOGA decision table.

### 3.2.2. Verification of the Application KB in PROLOGA

The system of decision tables is checked for intra- and inter- tabular anomalies: redundancy, ambivalence, circularity and deficiency that are described by 12 partial cases and are specified in (Vanthienen *et al.*, 1995). In the tabular verification method, inter-tabular anomalies are detected by comparing different parts of a system of decision tables against each other, because these anomalies result from interactions between two or more tables (Vanthienen *et al.*, 1997).

It is necessary to note the difference between an anomaly and error. The anomaly indicates the existence of a possible error. When a verification report is generated by PROLOGA, it should be decided according to a conceptual model whether the detected anomaly is an error, or it is not.

### 3.3. Functional Validation of the Application KB

Functional validation of the application KB is carried out using the expert system in CLIPS. It is built by combining already verified application KB with a KB of validated properties and validation method.

#### 3.3.1. KB of Validated Properties and Validation Method and its Join with Application KB

In our approach, we define the KB of validated properties and validation method that is implemented in CLIPS. It may be used for various kinds of applications with minor adaptations. The KB VPVM stores knowledge in the form of production rules. The instances of dynamic properties whose validation is implemented in the KB VPVM are the absence of static deadlocks, final state reachability, boundedness, and completeness. However, it is possible to define a system invariant property in the KB VPVM.

According to the review of validation methods made in (Pranevičius, 1990), the reachable state validation method was chosen. In a view of analysis of the execution paths, this method is similar to functional validation method suggested by Preece *et al.* (1996), that analyses the sequences of rules that must fire to achieve a goal.

Since both the KB VPVM and the application KB are written using CLIPS syntax, the needed adaptation changes are minor – they are an adaptation of description of validated properties for a specific application. For instance, in order to check the absence of the static deadlock property, the adaptation includes definition of specific continuous state coordinates; in order to check boundedness property, individual bounds on discrete state component coordinates have to be defined. Having combined the KB VPVM with the application KB, the expert system in CLIPS is built.

#### 3.3.2. Validation of the Application KB Using the Expert System in CLIPS

The expert system in CLIPS consists of the combined KB VPVM and the application KB, and CLIPS inference engine that is based on the forward chaining strategy. In order to perform the functional validation using the expert system, the initial and the final states of an analysed application are defined. An application model operates according

to the reachable states method. If the validated properties are violated, the expert system generates a corresponding report and a designer corrects the application KB accordingly. The next section describes the main phase of ESTELLE/Ag specification formation.

#### 4. Forming the Specification

In the first step, specification framework is constructed during a session with the specification editor PRAXIS (Pranevičius *et al.*, 1994) using the knowledge extracted from the validated and verified application KB. Note that while working with PRAXIS editor one has to use the same names of the concepts that have been used in the application KB.

In the second step, the generated specification framework is extended by adding with the behavioural description taken from the application KB. The supplementation process consists of the following phases:

- Finding the rules in the KB that correspond to ESTELLE/Ag specification constructions. These constructions describe either initial state of an aggregate, or condition of the aggregate state change due to an internal or an external event. Mapping between the rules in the application KB and ESTELLE/Ag specification constructions is presented in Table 1;
- Transformation of antecedents and consequents of the found rules to ESTELLE/Ag operators. The corresponding mapping has been defined too. A part of it is given

Table 1  
Mapping between rules in application KB and ESTELLE/Ag specification constructions

<i>Facts and production rules in application KB</i>	<i>Specification constructions</i>
Production rule containing (Initialize) in its LHS	Initialize begin ... end;
Production rule(s) containing (EndOfOperation <i>continuouscoordinate</i> ) in its LHS	when eop. <i>continuouscoordinate</i> begin ... end;
Production rule(s) containing (ArrivalOf <i>signal</i> ), (InputIp <i>ip</i> ), (ElementaryLink <i>linkname point-id</i> ) in its LHS	when <i>iip.point-id</i> begin ... end;

Table 2  
Fragment of mapping between facts in RHS of rules in application KB and ESTELLE/Ag operators

<i>Facts in rule consequent</i>	<i>ESTELLE/Ag operators</i>
(OutputSignal <i>signal component(s)</i> ), (OutputIp <i>oip</i> )	Output <i>oip.linkname(component(s))</i> ;
(bind ? <i>operation on</i> )	Start <i>operation</i> ;
(QueueOperation <i>item put</i> )	ENQ ( <i>item</i> );



in Table 2;

- Supplementation of specification constructions with the operators resulted from the previous phase.

A fragment of the created ESTELLE/Ag specification of the QS example using our technique is presented below. The description of the QS behaviour in case of arrival of an input signal is included in the rules  $r1$  and  $r2$ . These rules have been used to supplement when *qip.point-id* construction of ESTELLE/Ag specification framework. The PRAXIS generated code is written in regular font, the added code – in italic.

```

Trans
  when qip.point-id
    begin if request_processing="off" and query_size= 0 then
      begin Start request_processing;
      end;
      if request_processing="on" or query_size> 0 then
      begin ENQ("request");
      query_size:=query_size+1;
      end;
    end;
end;

```

## 5. Conclusions

Summarising the paper we would like to emphasise the following:

- The use of KB defines the creation of ESTELLE/Ag specifications. The production rules representation technique combined with PLA model is used for formal description of application structure and behaviour. The formalised description is presented at the knowledge level that permits to operate not using strict mathematic notation but the concepts of the application domain;
- KB validation and verification permits to perform analysis of the specification under creation at its initial construction stage.

## References

- Bruynooghe, M., N. Pelov and M. Denecker (1999). Towards a more declarative language for solving finite domain problems. In K. Apt, A. Kakas, E. Monfroy and F. Rossi (Eds.), *Proc. of the 1999 ERCIM/COMPULOG Workshop on Constraints*. pp. 1–14.
- D'Alameida, J., R. Achuthan, T. Radhakrishnam and V.S. Alagar (1992). Transformation of semi-formal specifications to VDM. In *Proc. of the Conference Knowledge-Based Software Engineering*. pp. 40–49.
- Facchi, C., M. Haubner and U. Hinkel (1996). *The SDL Specification of the Sliding Window Protocol Revisited. Technical Report TUM-19614*, Technische Univerit at Munchen.
- Fuchs, N.E., R. Schwitter (1996). Attempto Controlled English (ACE). In *Proc. of the CLAW 96, The First International Workshop on Controlled Language Applications*, Katholieke Universitet, Leuven, 26–27 March, 1996.

- Giarratano, J.C. (1993). *CLIPS Reference Manual. Basic Programming Guide, CLIPS Version 6.0*. Software Technology Branch, Lyndon B. Johnson Space Center, JSC-25012.
- Gupta, U.G. (1990). *Validating and Verifying Knowledge-based Systems*. IEEE Press, Los Alamitos, CA.
- Johnson, W.L., M.S. Feather and D.R. Harris (1991). The KBSA Requirements/Specification Facet: ARIES. In *Proc. of the Conference Knowledge-Based Software Engineering*. pp. 48–56.
- Meseguer, P., and A. Preece (1996). Assessing the role of formal specifications in verification and validation of knowledge based systems. In *Proc. 3rd IFIP International Conference on Achieving Quality in Software (AQuIS'96)*. Chapman and Hall. pp. 317–328.
- Mi, P., W. Scacchi (1995). A knowledge-based environment for modeling and simulating software engineering processes. *IEEE Trans. on Knowledge and Data Engineering*, **2**(3), 283–294.
- Miller, L., J. Hayes and S. Mirsky (1993). *Evaluation of Knowledge Base Certification Methods*. SAIC Report for U.S. Nuclear Regulatory Commission and electrical Power Research Institute NUREG/CR-6316 SAIC-95/1028, Vol. 4.
- Noah, S.A., M. Williams (2000). Exploring and validating the contributions of real-world knowledge to the diagnostic performance of automated database design tools. In P. Alexander and P. Flener (Eds.), *Proc. of ASE-2000: the 15th IEEE Conference on Automated Software Engineering*. IEEE CS Press. Grenoble, France.
- Pranevičius, H. (1990). Correctness analysis and simulation of computer network protocols by aggregate approach. *Informatica*, **1**(1).
- Pranevičius, H., V. Pilkauskas and A. Chmieliauskas (1994). *Aggregate Approach for Specification and Analysis of Computer Network Protocols*. Technologija, Kaunas.
- Preece, A. (2001). Evaluating verification and validation methods in knowledge engineering. In R. Roy (Ed.), *Micro-Level Knowledge Management*, Morgan-Kaufman. pp. 123–145.
- Preece, A., C. Grossner and T. Radhakrishnan (1996). Validating dynamic properties of rule-based systems. *International Journal of Human-Computer Studies*, **44**, 145–169.
- Preece, A., R. Shinghal (1994). Foundation and application of knowledge base verification. *International Journal of Intelligent Systems*, **9**, 683–702.
- Preece, A., R. Shinghar and A. Batarekh (1992). Principles and practice in verifying rule-based systems. *Knowledge Engineering Review*, **7**(2), 115–141.
- Russel, S., and P. Norvig (1995). *Artificial Intelligence – a Modern Approach*. Prentice Hall, Inc.
- Schreiber, A.Th., B. J. Wielinga and J.A. Breuker (1993). *KADS: A Principled Approach to Knowledge-Based System Development*. Academic Press, London.
- Sellini, F., C. Vargas and P.-A. Yvars (1997). Considerations about validation of knowledge models in KBE systems. In J. Vanthienen, F.V. Harmelen (Eds.), *Proc. 4th European Symposium on the Validation and Verification of Knowledge Based Systems*. Katholieke Universiteit Leuven. pp. 83–93.
- Sen, M., J. Minambres, A. Garrido and A. Almansa (2002). Logical formal description of expert systems. *Informatica*, **13**(2), 177–208.
- Specht, G. (1998). O!-LOLA – extending the deductive database system lola by object-oriented logic programming. *Informatica*, **9**(1), 107–118.
- Spirakis, P., B. Tampakas, K. Antonis, K. Hatzis and G. Pentaris (1996). *Specification Languages of Distributed and Communication Systems: State of the Art*. ESPRIT Long Term Research project Nr. 20244 report.
- Vanthienen J., G. Wets (1995). Integration of the decision table formalism with a relational database environment. *Information Systems*, **20**(7), 595–616.
- Vanthienen, J. (2000). *Prologa v.5 User's Manual*. Katholieke Universiteit Leuven.
- Vanthienen, J., C. Mues and G. Wets (1997). Inter-tabular verification in an interactive environment. In J. Vanthienen, F.v. Harmelen (Eds.), *Proc. 4th European Symposium on the Validation and Verification of Knowledge Based Systems*. Katholieke Universiteit Leuven. pp. 155–165.
- Velde, W.V., A. Aamodt (1994). Machine learning issues in CommonKADS. KADS-II/TII.4.3/TR/VUB/002/3.0.

**H. Pranevičius** is the head of the Business Informatics Department at the Kaunas University of Technology and the research group “Formal Specification, Verification and Simulation of Distributed Systems.” The research group has been working in the field of creation of formal description methods for systems with distributed information processing and for modelling of complex systems for a long time. The results of investigations have been successfully applied when creating the computerised system for specification, validation, and simulation/modelling of computer network protocols. H. Pranevičius has published over 200 scientific papers and 4 monographs.

**G. Budnikas** is PhD student, a member of prof. H. Pranevičius’ research group. His research interests include creation of formal specifications using knowledge engineering techniques, validation, and verification of knowledge bases.

**ESTELLE/Ag specifikacijų sudarymas panaudojant žinių bazes**

Henrikas PRANEVIČIUS, Germanas BUDNIKAS

Straipnyje pateikiama metodika, kuri apibrėžia ESTELLE/Ag specifikacijų sudarymą naudojant žinių bazes (ŽB). Probleminės srities ŽB sudaroma naudojant žinių išgavimo metodiką orientuotą į atkarpomis tiesinių agregatų modelių sukūrimą. Probleminės srities produkcinės taisyklės transformuojamos į sprendimo lenteles ir ŽB statinės savybės tikrinamos sistemoje PROLOGA. Toliau, probleminės srities ŽB apjungiami su apibrėžta validuojamų savybių ir validavimo metodo ŽB. Ir probleminės srities žinių bazės dinaminės savybės tikrinamos ekspertinėje sistemoje CLIPS aplinkoje. Validuota probleminės srities ŽB naudojama apibrėžiant ESTELLE/Ag specifikacijos struktūrą PRAXIS redaktoriaus pagalba ir papildant PRAXIS sugeneruota struktūrą probleminės srities funkcinio aprašu. Metodika iliustruojama vienkanalės aptarnavimo sistemos pavyzdžiu.