TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
Baltic Journal on Sustainability
2008
14(3): 328–343

PLA-BASED FORMALIZATION OF BUSINESS RULES AND THEIR ANALYSIS BY MEANS OF KNOWLEDGE-BASED TECHNIQUES

Henrikas Pranevičius[1], **Germanas Budnikas**[2]

Business Informatics Department, Kaunas University of Technology,
Studentų g. 56–301, 51424 Kaunas, Lithuania, Telephone/Fax: +370 37 451654
E-mails: [1]henrikas.pranevicius@ktu.lt; [2]germanas.budnikas@ktu.lt

Received 6 April 2008; accepted 5 September 2008

Abstract. The paper presents an approach that applies the knowledge engineering techniques for representing and analysing business rules. These rules are represented by production rules using concepts of a state-based piece-linear aggregate (PLA) model. A knowledge base of the business rules is analysed by checking its consistency (static properties) and dynamic constraints (dynamic properties). The analysis is performed by applying methods for decision table verification and reachable state validation as well as supporting tools. The proposed approach is illustrated by an example of Userv insurance company services.

Keywords: business rules, PLA, knowledge base, decision table, consistency and dynamic constraints.

Reference to this paper should be made as follows: Pranevičius, H.; Budnikas, G. 2008. PLA-based formalization of business rules and their analysis by means of knowledge-based techniques, *Technological and Economic Development of Economy* 14(3): 328–343.

1. Introduction

Business rules are significant for enterprises because they define how they are doing business. They are also popular in the information system community, as they are able to make applications flexible. One of the most important parts in the development of information systems is problem representation and analysis of domain descriptions, which, in our case, are expressed in a form of business rules. Possible inconsistencies like redundancies, contradictions or missing rules as well as dynamic failures like inability to reach a defined goal should be corrected as early as possible to minimise the cost of development. Checking for consistency as well as dynamic constraints is emphasised in formal methods and formal specifications. The most popular formal specification languages used for describing the systems are SDL, Lotos, Estelle, PLA (Facchi *et al.* 1996), (Spirakis *et al.* 1996), (Pranevičius 1991, 2005). Checking the business rules consistency is highly important. Examples of related works could be Spreeuwenberg 2003; Halle 2001; Date 2000; Baisley 2004. Use of models (especially formal ones) while representing and analysing business rules is propagated in many works, e.g. (Bajec and Krisper 2005; Pranevičius 2003), as it simplifies the rule formalisation and ensures a higher clarity and consistency of rule descriptions.

PLA is a formal state-based model (Pranevicius 1991, 2005). It is successfully used for formal specification, validation and simulation of complex systems including computer network protocols, transport systems, medical applications, business processes, etc. for several decades. Use of PLA model in representing and analysing business rules should give us an advantage of applying a formal model as well as accompanying analysis techniques. In this paper we will explore the possibilities to use PLA and related techniques for business rules.

Business experts find difficult to understand formal specification languages. Many authors (e.g. Bruynooghe *et al.* 1999) believe that the declarative style of description that is used in knowledge bases is more understandable and acceptable than the procedural style (the latter is used in PLA formal specification language). This is because a problem is described at the knowledge level "at which the knowledge engineer specifies expertise" (Velde and Aamodt 1994). Representation of knowledge used in knowledge-based systems (KBS) is close to the way of thinking by a human. Conditional and sequential parts of the most business rules correspond to a structure of a production rule representation. In this way, representation of business rules by production rules is natural.

State structure, conditions for state changes due to incoming requests and internal events are strictly defined in the PLA model. Adding the PLA model to the representation process assigns extra value in terms of easier to represent a structure of business processes expressed using business rules, their interaction; easier to perform checking.

Our approach for representing business rules, which are outlined in section 2, is based on production rule representation and PLA model elements. This approach is explained in section 3. Business rules consistency is expressed using general properties – non-redundancy, non-conflictness, non-deficiency. The consistency of business rules is checked by transforming PLA productions to decision tables (DT) and applying tabular static verification technique that is implemented in Prologa system. The consistency checking technique is described in section 4. Dynamic constraints of business rules are expressed using general dynamic properties – absence of static deadlocks, final state reachability, boundedness and completeness. The constraints are checked using an expert system that is built by combining PLA KB with knowledge base of dynamic properties and validation method. The technique is presented in section 5. Section 6 illustrates the proposed technique with the example of business rules describing Userv company insurance services (BRForum 2005). Related works are outlined in section 7. Conclusion sums up the paper.

2. Business rules and business processes

Business rules are explicit statements that regulate how a business operates and how it is structured. In (Kovacic 2004) a business process is defined as a subset of business activities performed by the organisation to achieve the goals. Activities correspond to different stages of process execution, which in their turn, can be started by influence of some events. Business processes can be viewed as a sequence of business rules that define how the flow of control is passed from one activity to another and under what conditions the transition can happen.

3. Business rules representation using concepts of PLA model

According to the review made in Halle (2001), classification categories of BR include the following: rules describing situations (e.g. state changes due to business events), constraints, facts asserting structure, terms corresponding to actors, events, parameter values, actions, etc. These concepts and rules already present in the state-based formal piece linear aggregate PLA model (Pranevičius 1991, 2005). The model defines notions that can be successfully applied for business rule representation and analysis. Input and output requests, their structure, internal and external (business) events along with their cause, activities, computational parameters, state structure and rules describing how state is changed, actors and their interconnection, are all defined in the formal PLA model. Representation of business rules in a declarative way – by using productions, is natural and will be used in our approach.

Therefore, we propose for each rule category to use an appropriate rule template, which can be seen as a sentence pattern that tells how to describe the rules that belong to a particular category. Further (Table 1), we present examples of business rule categories and how they are represented by rule templates expressed in a form of predicates and productions of a knowledge base based on the PLA model (PLA KB).

The representation using PLA KB is performed by putting knowledge of business rules to corresponding elements of PLA KB, which form and structure are defined with respect to formal PLA model. Such a way of representing the business rules by putting their knowledge to the constructs of the defined structure is suggested in Bajec and Krisper (2005), as it simplifies the rule formalisation and ensures higher clarity and consistency of the rule descriptions.

After creation of the PLA KB, it is checked for consistency and dynamic failures by analysing its general static and dynamic properties.

4. Checking the business rules consistency

Business rules consistency means that the business rules are expressed in the right way. Checking this property is known as verification (Spreeuwenberg 2003). Consistency of business rules usually is expressed in terms of non-redundancy, non-conflictness, non-deficiency (Halle 2001). These constraints by their nature correspond to the ones that are analysed while checking static properties (i.e. performing static verification) of rule bases.

There are a lot of works devoted for checking business rule consistency, e.g. Alagar and Periyasamy (2002), Ouyang *et al.* (2005), Spreeuwenberg (2003), Pranevičius and Misevičienė (2008), Rouached *et al.* (2006), Goedertier and Vanthienen (2005). Most of them propagate static analysis. Background theories of these techniques include event calculus, Petri nets, model checking, and decision tables.

A decision table consists of 4 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.

Table 1. Examples of business rule categories and their representation by PLA KB

Business rule category	PLA KB predicate or production
incoming request	IncomingRequest ($a_i, x_j^1, \ldots, x_j^{c_{ic}^i}$), where a_i – symbolic name of the *i*th actor; $x_j^1, \ldots, x_j^{c_{ic}^i}$ – components of the request.
actor’s state	State ($a_i, w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$), where $w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$, are actions and computation parameters describing the state.
state change and output of the request after occurrence of an external event	IF IncomingRequest ($a_i, x_j^1, \ldots, x_j^{c_{dc}^i}$) and State ($a_i, w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$) and Aux ($a_i, w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$) THEN State ($a_i, w_i^1*, \ldots, w_i^{c_{cc}^i}*, d_i^1*, \ldots, d_i^{c_{dc}^i}*$) and OutcomingRequest ($a_i, y_j^1, \ldots, y_j^{c_{oc}^j}$) where predicate Aux describes auxiliary conditions that check state values.
state change and output of the request after occurrence of an internal event	IF EndOfOperation (a_i, w_i^u) and State ($a_i, w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$) and Aux ($a_i, w_i^1, \ldots, w_i^{c_{cc}^i}, d_i^1, \ldots, d_i^{c_{dc}^i}$) THEN State ($a_i, w_i^1*, \ldots, w_i^{c_{cc}^i}*, d_i^1*, \ldots, d_i^{c_{dc}^i}*$) and OutcomingRequest ($a_i, y_j^1, \ldots, y_j^{c_{oc}^j}$)

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. The decision table DT is defined (Vanthienen *et al.* 1997) in a following way:

$$CS_1 \times CS_2 \times \ldots \times CS_m \rightarrow AV_1 \times AV_2 \times \ldots \times AV_n,$$

where CS_i is a set of condition states, AV_j – set of action values.

Business rules in our approach are represented by production rules of PLA KB. Most of inconsistencies in rule-based systems can be solved using DT (Vanthienen 2000) and the tabular verification technique is computerised in Prologa system (Vanthienen *et al.* 1997), our approach employs these facilities. Therefore, in order to perform checking of consistency of business rules represented in PLA KB, its production rules have to be transformed to Prologa DTs. As stated in Vanthienen (2000), a DT is equivalent to a set of production rules. Consistency constraints for DTs have direct correspondences to the ones, defined for rules, which make the PLA KB.

The transformation to Prologa DTs is specific with respect to the PLA model and employs some of its concepts that have been mentioned in the beginning of this section. Production rules are transformed to the DTs of certain groups, thus enabling to fully exploit advantages of tabular representation to perform consistency checking. Further, we present an outline of suggested steps of transformation of PLA KB productions to single hit DTs.

1. Productions describing certain types of events are transformed to corresponding event tables;
2. Predicates of antecedent (consequent) part of a rule are written in a form of condition (action) subjects in a DT;
3. Decrease of a computational parameter by a constant value *Const* is marked with not ($d_j^i = d_j^i + Const$). This notation is used while non-conflictness property is being checked.

Static verification technique for analysis of PLA KB is based on works of Vanthienen (2000), Vanthienen *et al.* (1997). Further we present a portion of this technique. A DT contains a *subsumed column pair*, if and only if it includes a pair of columns (CS_j; AS_j), (CS_k; AS_k) ($1 \le j, k \le t$), $j \ne k$ for which: $CS_j \subseteq S_k$ and $AS_j \supseteq AS_k$ (Mues 2002).

A *subsumed column pair* in a single hit DT is represented by single column that corresponds to several PLA KB productions.

Thus, the subsumed column pair is detected if several PLA KB productions correspond to the same DT column. Full description of the PLA KB static verification technique as well as transformation steps of PLA KB constructs to single hit DTs are presented in Budnikas (2004), Pranevičius and Budnikas (2003b).

5. Checking dynamic constraints of business rules

The dynamic constraints (or 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).

Functional validation of the PLA KB is carried out using the dynamic validation expert system (DVES) in CLIPS. CLIPS – C Language Integrated Production System is a tool for productive development and delivery of expert systems (CLIPS 2003). The expert system is built by joining:

- statically verified PLA KB with

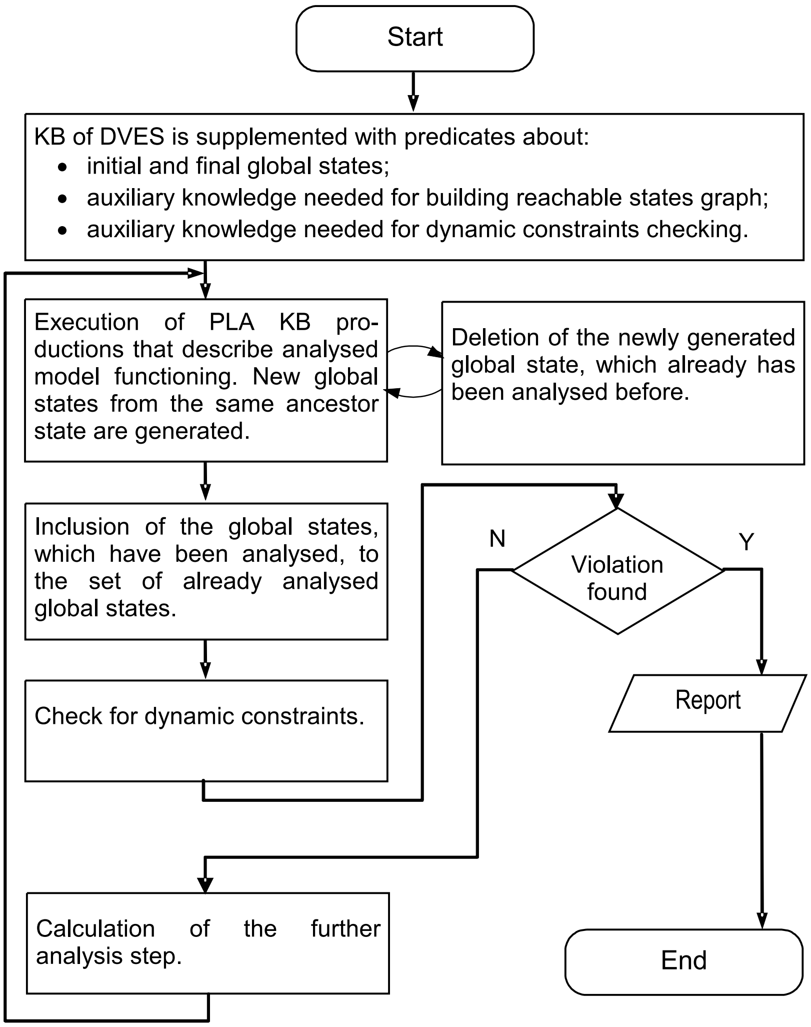

Start
KB of DVES is supplemented with predicates about:
• initial and final global states;
• auxiliary knowledge needed for building reachable states graph;
• auxiliary knowledge needed for dynamic constraints checking.
Execution of PLA KB productions that describe analysed model functioning. New global states from the same ancestor state are generated.
Deletion of the newly generated global state, which already has been analysed before.
Inclusion of the global states, which have been analysed, to the set of already analysed global states.
Check for dynamic constraints.
N
Violation found
Y
Report
Calculation of the further analysis step.
End

Fig. 1. Dynamic validation scheme

- knowledge base of dynamic properties and validation method (KB DPVM) (Pranevičius and Budnikas 2003a).

The defined KB DPVM, which is implemented in CLIPS, may be used for various kinds of applications. The instances of general dynamic properties are the absence of static deadlocks, final state reachability, boundedness of countable state parameters and completeness.

While joining PLA KB with KB DPVM, the general descriptions of dynamic properties are adjusted with respect to actual business process constraints. For instance, in order to check the boundedness property, individual bounds are being defined.

Reachable states method and forward chaining mechanism are used in the dynamic validation expert system. Reachable states method provides a schema for the generation of all

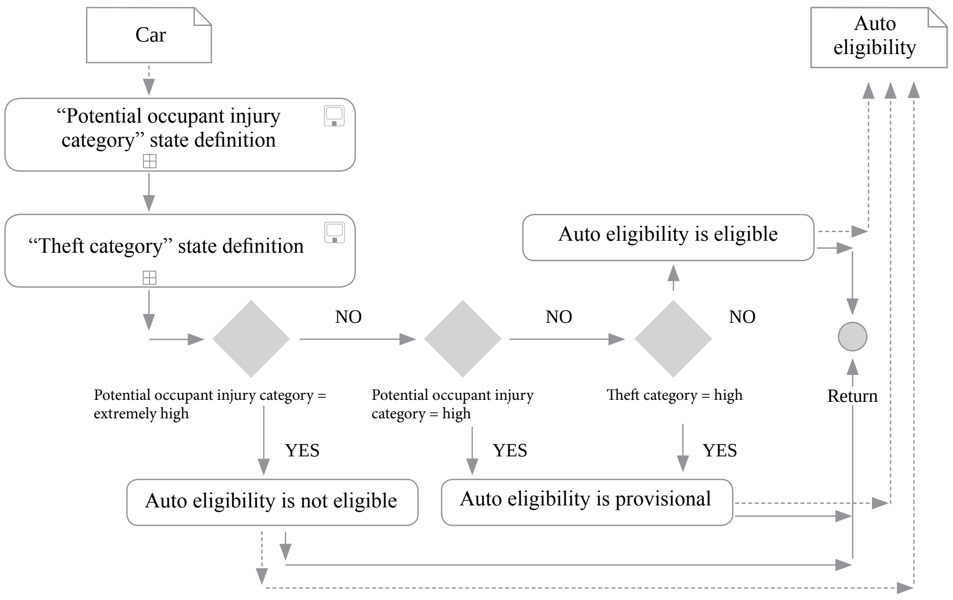

Car
"Potential occupant injury category" state definition
"Theft category" state definition
NO
NO
NO
Potential occupant injury category = extremely high
Potential occupant injury category = high
Theft category = high
YES
YES
YES
Auto eligibility is not eligible
Auto eligibility is provisional
Auto eligibility is eligible
Auto eligibility
Return

Fig. 2. Business process of definition of automobile eligibility

possible model execution paths while analysing impacts on global states[1] 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. 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.

Fig. 1 illustrates the process of dynamic constraints checking by the reachable states method. Having combined the KB DPVM with the PLA KB, the expert system (ES) in CLIPS is built.

In order to perform the functional validation using the ES, the initial and the final states of an analysed business process are defined. Business rules are executed according to the reachable states method. If the validated dynamic constraints are violated, the expert system generates a corresponding report and a designer corrects the KB accordingly.

6. Illustration of the proposed technique

To illustrate the techniques we employ an example of business rules presented in BRForum (2005). Userv company provides insurance services. Business rules manage risk factors and address client segmentation, eligibility, pricing and cancellation policies. In this paper we present a part of eligibility business rules and their analysis by our approach.

[1] Global model state is composed using state parameters of all aggregates of the analysed model.

Business processes and business rules

We present business processes describing automobile eligibility (Fig. 2), And sub-processes that describe:

- potential Theft Category definition (Fig. 3);
- potential occupant injury category definition.

Business process describing potential theft category definition, correspond to business rules as stated in (BRForum 2005):

If the car is convertible, then the car's potential theft rating is high.
If the car's price is greater than $45,000, then the car's potential theft rating is high.
If the car model is on the list of "High Theft Probability Auto", then the car's potential theft rating is high.
If all of the following are true, then the car's potential theft rating is moderate. • car's price is between $20,000 and $45,000, • car model is not on the list of "High Theft Probability Auto"
If all of the following are true, then the car's potential theft rating is low: • car's price is less than $20,000 • car model is not on the list of "High Theft Probability Auto"

Representation of business rules – development of PLA KB

Actors (aggregates) for this model are:

Aggregate (Customer), and Aggregate (Userv)

Customer aggregate is described by continuous state parameters (operations):

State(Customer, request_formation)

Userv aggregate is described by continuous state parameters as well as discrete state parameters:

State (*Userv, theft_definition,*
injury_definition,
auto_eligibility_definition, driver_category_definition,
eligibility_scoring_wrt_auto,
eligibility_scoring_wrt_driver,
eligibility_scoring_wrt_client,
eligibility4insurance_defnition,
auto_premiums_wrt_type,
auto_premiums_wrt_age,
auto_premiums_wrt_medical_coverage,
auto_premiums_wrt_safety, auto_premiums_wrt_theft,
discounts_wrt_safety,
discounts_wrt_theft,
driver_premiums_definition,
driver_premiums_wrt_incedents,
driver_premiums_wrt_driver_type,

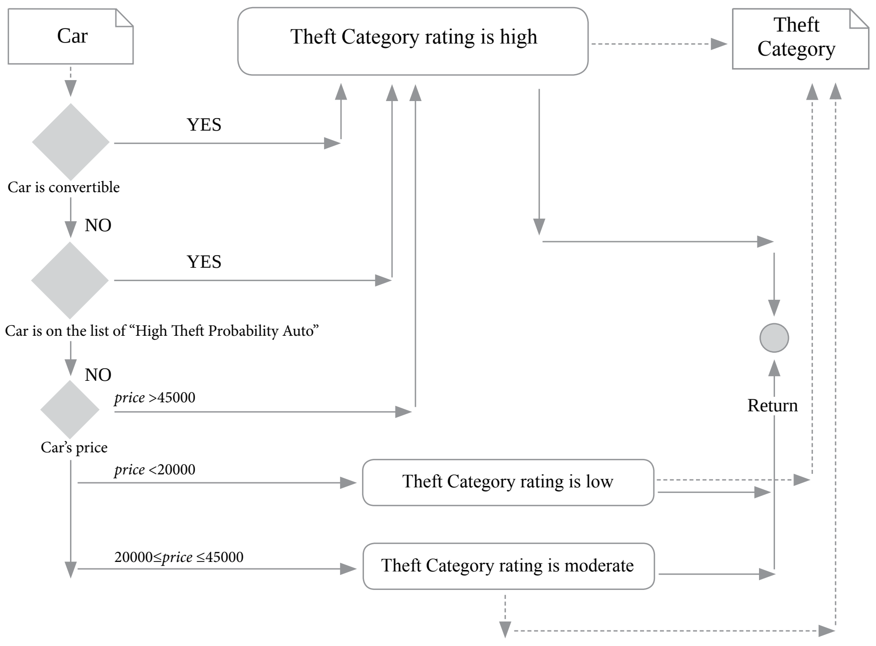

Car
Theft Category rating is high
Theft Category
YES
Car is convertible
NO
YES
Car is on the list of "High Theft Probability Auto"
NO
price >45000
Car's price
Return
price <20000
Theft Category rating is low
20000≤price ≤45000
Theft Category rating is moderate

Fig. 3. Business process of Potential Theft Category definition

decision_making, theft_rating, price,
hight_theft_probability_auto, convertible_car,
client_type, eligibility_score)

For the briefness while referencing State predicate we will list terms that are related to the certain rule.

R1:

IF **End Of Operation** (Userv, *Theft_definition*) and
State (*Userv, theft_definition, injury_definition,*
theft_rating, ..., price, hight_theft_probability_ auto, convertible_car)
and *price* < 20000 and
hight_theft_probability_auto = **False**

THEN
*theft_rating** = **Low** and
*injury_category_definition** = **Active** and
State (*Userv, injury_category_definition**,
*theft_rating**, ...)

R2:

IF **End Of Operation** (Userv, *Theft _definition*) and

```
        State    (Userv, injury_definition, theft_rating, ..., price,
                 hight_theft_probability_auto, convertible_car)
                 and
                 20000 <= price <= 45000 and
                 hight_theft_probability_auto = False
        THEN
                 theft_rating* = Moderate and
                 injury_category_definition* = Active and
        State    (Userv, injury_category_definition*,
                 theft_rating*, ...)
    R3:
        IF       End Of Operation (Userv, Theft_definition) and
        State    (Userv, injury_definition, theft_rating, ...,
                 price, hight_theft_probability_auto,
                 convertible_car) and
                 (price > 45000 or
                 hight_theft_probability_auto = False or
                 convertible_car = True)
        THEN
                 theft_rating* = Moderate and
                 injury deifnition* = Active and
        State    (Userv, injury_definition*, theft_rating*, ...)
```

Checking the business rules – static analysis of Userv PLA KB

Predicates and productions of PLA KB are represented by PROLOGA decision tables. Next we present a decision table corresponding our rules (see Fig. 4).

While verifying set of decision tables, we can notice that 1, 2 and 3 action rows in 2, 4 and 5 columns represent ambivalence anomaly – different values (Low/High, Moderate/High) are assigned for the same set of conditions. To fix this, productions R1 and R2 of PLA KB are corrected – their condition parts are supplemented with additional condition:

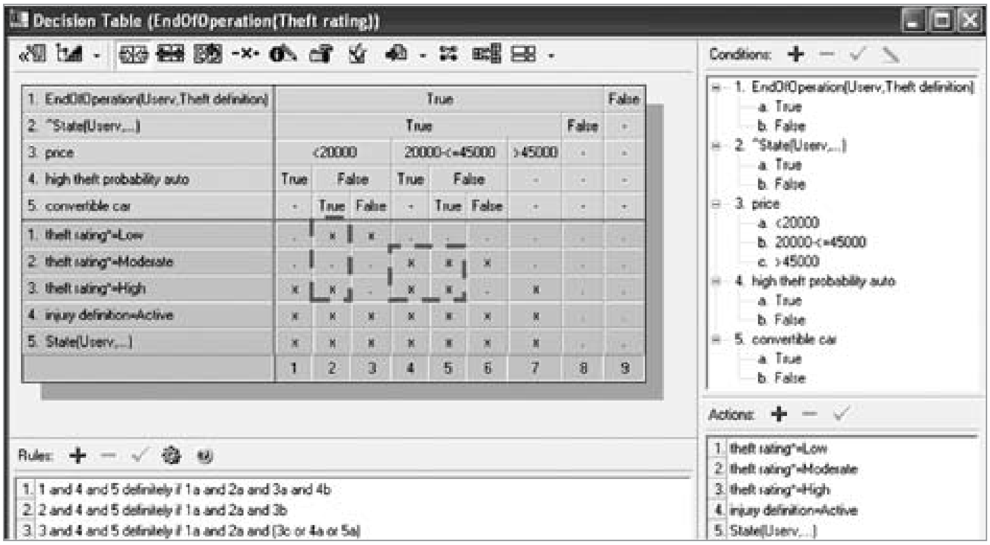
convertible_car = **False**.

Checking the business rules – dynamic analysis of Userv PLA KB

To illustrate how dynamic analysis is performed we present here description of some properties for the analysed example.

Absence of static deadlocks property

```
        IF       GlobalState(parrent_state, current_state,
                 request_formation,
                 theft_definition,
                 injury_definition,
                 auto_eligibility_definition,
                 driver_category_definition,
                 eligibility_scoring_wrt_auto_definition,
```


Decision Table (EndOfOperation(Theft rating))

1. EndOfOperation(Userv,Theft definition)	True								False
2. ^State(Userv,...)	True							False	-
3. price	<20000			20000-<=45000			>45000	-	-
4. high theft probability auto	True	False		True	False		-	-	-
5. convertible car	-	True	False	-	True	False	-	-	-
1. theft rating^=Low	.	x	x	.	.	.	.	.	.
2. theft rating^=Moderate	.	.	.	x	x	x	.	.	.
3. theft rating^=High	x	x	.	x	x	.	x	.	.
4. injury definition=Active	x	x	x	x	x	x	x	.	.
5. State(Userv,...)	x	x	x	x	x	x	x	.	.
	1	2	3	4	5	6	7	8	9

Rules:

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

Conditions:

1. EndOfOperation(Userv,Theft definition)
   a. True
   b. False
2. ^State(Userv,...)
   a. True
   b. False
3. price
   a. <20000
   b. 20000-<=45000
   c. >45000
4. high theft probability auto
   a. True
   b. False
5. convertible car
   a. True
   b. False

Actions:

1. theft rating^=Low
2. theft rating^=Moderate
3. theft rating^=High
4. injury definition=Active
5. State(Userv,...)

Fig. 4. Static verification in PROLOGA

eligibility_scoring_wrt_driver_definition,
eligibility_scoring_wrt_client_definition,
eligibility4insurance_defnition,
auto_premiums_wrt_type_definition,
auto_premiums_wrt_age_definition,
auto_premiums_wrt_medical_coverage_definition,
auto_premiums_wrt_safety_definition,
auto_premiums_wrt_theft_definition,
discounts_wrt_safety_definition,
discounts_wrt_theft_definition,
driver_premiums_definition,
driver_premiums_wrt_incedents_definition,
driver_premiums_wrt_driver_type_definition,
decision_making, theft_rating,
price, hight_theft_probability_auto, convertible_car,
client_type, eligibility_score)
and *request_formation* = **Passive**
and *theft_definition* = **Passive**
and …
and *driver_premiums_wrt_driver_type_definition* = **Passive**

THEN **Validation_Report** (*'Deadlock present in state' & current_state & 'which is originated from' & parrent_state*) and **StopInference**

Final state reachability

IF **GlobalState** (*parrent_state, current_state*, Active, Passive, Passive, Passive,

Passive, Passive, Passive, Passive, Passive, Passive,Passive, Passive, Passive, Passive, Passive, Passive, Passive,Passive, Passive, Active, Moderate, 50000, *hight_theft_probability_ auto, convertible_car, client_type, eligibility_ score*)
THEN **Validation_Report** (*'Final state achieved'*) and **StopInference**

Boundedness of countable state parameters

IF **GlobalState**(*parrent_state, current_state, ...*)
and price < 0
THEN **Validation_Report** (*'Boundedness constraint violated in state' & current_state & 'which is originated from' & parrent_state*) and **StopInference**

These productions are executed along with generation of every new global state of the reachable state graph.

7. Related works

Alagar and Periyasamy (2001) suggest formal specification formalism for security and business policies and workflow schemes that uses Hoare style axiomatic verification approach for detecting conflicts and proving security of business transactions. In the specification formalism a workflow is specified as an extended state machine having the syntax of state charts. Individual security and business policies are specified in set theory and logic. Semantics for policy updates is given for every state of the state machine description of a workflow. Verification procedure is represented as a proof of correctness criteria in each state of the state machine. Our approach is similar to this one by the idea of state machine using as a background for formal specification and proving correctness criteria in each state of the state machine while checking dynamic constraints of an analysed business process. However our approach checks for general dynamic constraints as well as additionally performs static analysis of general properties in contrast to the compared source.

Ouyang *et al.* (2005) presents a tool for automated analysis of BPEL (Business Process Execution Language) processes. The analysis is performed by translating BPEL processes to Petri nets and applying existing Petri net analysis techniques. The tool statically verifies unreachable actions and conflicting message-consuming activities. Set of similar general properties is statically checked by using Ouyang *et al.* (2005) method, while we check these constraints dynamically, but the set of general static properties that is analysed by our approach is broader.

Rouached *et al.* (2006) proposes an event-based approach for checking consistency of business processes, for mining the business process events, and for analysing the process execution. The approach framework uses Event Calculus (a temporal formalism expressed in a set of predicate logic designed to model and reason about scenarios characterised by a set of events). Static analysis is executed by transforming BPEL constructs into EC predicates and model-checking business processes with respect to temporal constraints. Dynamic analysis is done by comparing logical predicates describing business process constraints with respect to the events that occur during the process execution Event-based approach is used as a background method for business process representation and consistency checking in our and (Rouached *et al.* 2006) approaches. Transformation to suitable representation model in order to perform static analysis is used in both approaches. In Rouached *et al.* (2006) the dynamic analysis is executed while running the model and comparing logical predicates of business constraints. In this way, all possible states are analysed. However, dynamic constraint checking in our method enables analysis of all possible states of an analysed business processes represented by business rules since reachable state analysis method is employed.

Alagar and Periyasamy (2001) proposes formal model and a specification language Business Transaction Object-Z to write business rules. However, in contrast to our approach, no static or dynamic constraint checking is used to analyse the developed description of business rules.

Pranevicius and Misevičienė (2008) work is very similar to ours technique – they propose the same PLA model for formalising business rules as well as using reachable state method for dynamic analysis of general and invariant properties. In constraint to this technique our approach proposes to use static analysis to check general inconsistencies of the business rules before running the model.

Rule manager (Rule manager 2007) checks for the same set of cases as our proposed technique. The following cases are being explored: contradictions, redundancies, and incompleteness. Bergeron *et al.* (2004) state that static and dynamic analyses are complementary. They propose to use static analysis first. In our work at the beginning of the checking we use static verification technique that complements the dynamic analysis of business rules. Our approach as well as many others, example of which is Goedertier and Vanthienen (2005), exploits advantages of tabular representation in order to perform verification by transforming certain representation structure to DTs. While performing static verification of PLA KB, we use results of Vanthienen *et al.* (1997), whereas when analysing the dynamic constraints 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.

8. Conclusions

In this paper we have presented a technique for representation and analysis of business rules. Using an illustrative example, we showed that PLA model can be successfully used as a background model for business rules representation. It gives us the following benefits:

- Tabular verification method can be successfully applied for analysis of static constraints of business rules expressed in PLA knowledge base. These constraints are non-redundant, non-conflicting, and non-deficient;
- PLA-related validation method for analysis of dynamic properties using reachable states graph can be successfully applied while checking general dynamic constraints of business rules, i.e. absence of static deadlocks, final state reachability, boundedness of countable state parameters, and completeness.

Application area of the approach proposed are business rules describing interactions between structural parts of a system or organisation. This statement is based on a fact that

PLA model is primarily suited for specification and analysis of interacting complex 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 Vanthienen, 2000, (CLIPS, 2003).

References

Alagar, V. S. and Periyasamy, K. 2001. BTOZ: A Formal Specification Language for Formalizing Business Transactions, *TOOLS* 39: 240–252.

Alagar, V. S. and Periyasamy, K. 2002. Specification and verification of secure business transaction systems, *Lecture notes in computer science* Vol. 2540, ed. W. I. Grosky, F. Plasil, 240–252.

Baisley, D. 2004. A Metamodel for Business Vocabulary and Rules: Object-Oriented Meets Fact-Oriented, *Business Rules Journal* 5(7).

Bajec, M.; Krisper, M. 2005. A methodology and tool support for managing business rules in organisations, *Information Systems* 30(6): 423–443.

Bergeron, J.; Debbabi, M.; Desharnais, J.; Erhioui, M.; Lavoie, Y.; Tawbi, N. 2004. Static detection of malicious code in executable programs, *International Journal of Requirement Engineering*: 1–9.

BRForum. 2005. Userv product derby case study. Technical report, Business Rules Forum.

Bruynooghe, M.; Pelov, N. and Denecker, M. 1999. Towards a more declarative language for solving finite domain problems, in *Proc. of the ERCIM/ COMPULOG Workshop on Constraints*, 1–14.

Budnikas, G. 2004. *Development and analysis of aggregate specifications using knowledge bases*. PhD dissertation. Kaunas University of Technology.

CLIPS Reference Manual. 2003. Basic Programming Guide.

Date, C. J. 2000. *What Not How: The business rules approach to application development*. Addison-Wesley Longman.

Facchi, C.; Haubner, M. and Hinkel, U. 1996. *The SDL Specification of the Sliding Window Protocol Revisited*. Technical Report TUM-I9614, Technische Universitat at Munchen.

Goedertier, S.; Vanthienen, J. 2005. Rule-based business process modeling and execution, in *Proc. of the International IEEE EDOC Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2005)*, ed. G. Guizzardi and G. Wagner, CTIT Workshop Proceeding Series, 67–74.

Halle, B. 2001. *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. John Wiley.

Kovacic, A. 2004. Business renovation: business rules (still) the missing link, *Business Process Management Journal* 10(2): 158–170.

Mues, C. 2002. *On the Use of Decision Tables and Diagrams in Knowledge Modeling and Verification*. PhD dissertation. Katholieke Universitet Leuven.

Ouyang, C.; Verbeek, H. M. W.; Aalst, W. M. P.; Breutel, S.; Dumas, M. and Hofstede, A. H. M. 2005. WofBPEL: A Tool for Automated Analysis of BPEL Processes, in *Proc. of the 3rd International Conference on Service Oriented Computing, Lecture Notes in Computer Science No. 3826*, Springer, 484–489.

Pranevičius, H. 1991. *Aggregate approach for specification, validation, simulation and implementation of computer network protocols, Lecture Notes in Computer Science No 502*: Springer-Verlag, 433–477.

Pranevičius, H. 2003. Formalization and simulation of business process, in *Proc. of the International Conference Modelling and Simulation of Business Systems*, Vilnius, May 13–14, Ed. H. Pranevicius, E. Zavadskas, B. Rapp. Kaunas: Technologija, 198–202.

Pranevičius, H. 2005. *Formal specification and analysis of computer network protocols.* Monograph. Technologija.

Pranevičius, H. and Budnikas, G. 2003a. Creation of Estelle/Ag Specifications Using Knowledge Bases, *Informatica* 14(1): 63–74.

Pranevicius, H. and Budnikas, G. 2003b. Static Verification of Aggregate Specifications, *Information Technology and Control* 4(29): 39–44.

Pranevičius, H. and Misevičienė, R. 2008. Verification of business rules using logic programming means, in *Proc. of the International Conference Modelling of Business Industrial and Transport Systems, Riga, Latvia, 7–10 May 2008*. Ed. E. Kopytov, H. Pranevičius, E. Zavadskas, I. Yatskiv. Riga: Publishing department of Transport and Telecommunication Institute, 99–106.

Preece, A. 2001. *Evaluating Verification and Validation Methods in Knowledge Engineering, Micro-Level Knowledge Management*, Morgan-Kaufman: 123–145.

Preece, A.; Grossner, C.; Radhakrishnan, T. 1996. Validating Dynamic Properties of Rule-Based Systems, *International Journal of Human-Computer Studies* 44: 145–169.

Rouached, M., Perrin, O. and Godart, C. 2006. Towards Formal Verification of Web Service Composition, in ed. S. Dustdar, J. L. Fiadeiro, and A. Sheth, *Lecture Notes in Computer Science* 4102: 257–273.

Rule Manager. 2007. User manual. Available from Internet: <*http://www.acumenbusiness.com/*>.

Spirakis, P.; Tampakas, B.; Antonis, K. Hatzis and K. Pentaris, G. 1996. *Specification Languages of Distributed and Communication Systems*: State of the Art, ESPRIT Long Term Research project Nr. 20244 report.

Spreeuwenberg, S. 2003. Using Verification and Validation Techniques for High-quality Business Rules, *Business Rules Journal* 4(2), URL: <*http://www.brcommunity.com/a2003/b132.html*>.

Vanthienen, J. 2000. *Prologa v.5 User's manual*, Katholieke Universiteit Leuven.

Vanthienen, J.; Mues, C.; Wets, G. 1997. Inter-tabular Verification in an Interactive Environment, in *Proc. of the 4th European Symposium on the Validation and Verification of Knowledge Based Systems*, Katholieke Universiteit Leuven: 155–165.

Velde, W. V. and Aamodt, A. 1994. *Machine learning issues in CommonKADS*, KADS-II/ TII.4.3/TR/ VUB/002/3.0.

PLA METODU GRĮSTAS VERSLO TAISYKLIŲ FORMALIZAVIMAS IR ANALIZĖ TAIKANT ŽINIOMIS GRĮSTAS METODIKAS

H. Pranevičius, G. Budnikas

Santrauka

Straipsnyje pateikiamas būdas, kurio remiantis taikomos žiniomis grįstos metodikos verslo taisyklėms pavaizduoti ir analizuoti. Verslo taisyklės iliustruojamos produkcinėmis taisyklėmis, naudojant būsenomis grįsto atkarpomis tiesinių agregatų (PLA) modelio konceptus. Verslo taisyklių žinių bazė analizuojama tikrinant jos suderinamumą (statines savybes) ir dinaminius suvaržymus (dinamines savybes). Analizė atliekama taikant sprendimo lentelių verifikavimo ir pasiekiamų būsenų validavimo metodus bei šiuos metodus realizuojančias programines priemones. Siūlomas būdas iliustruojams *Userv* draudimo kompanijos paslaugų pavyzdžiu.

Reikšminiai žodžiai: verslo taisyklės, PLA, žinių bazė, sprendimo lentelė, suderinamumo ir dinaminiai suvaržymai.

Prof. Dr. Habil. Henrikas PRANEVIČIUS works in the Business Informatics Department at the Kaunas University of Technology. He is a scientific leader of 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 complex systems for a long time. The results of investigations have been successfully applied 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.

Dr. Germanas BUDNIKAS is a member of Prof. H. Pranevičius' research group. His interests include creation of formal specifications using knowledge engineering techniques, validation, and verification of knowledge bases. The author of about 10 scientific papers on the topic of his research.