

Methods of Detection and Resolution of Anomalies in Business Rule: Comparative Study

Addamssiri Najiba LAVETE Laboratory Department of Mathematics and Computer Science, FSTS, Hassan 1st University Settat 26000	Azzeddine Dahbi LAVETE Laboratory, Department of Mathematics and Computer Science, FSTS, Hassan 1st University Settat 26000	Mohammed Mouhir LAVETE Laboratory, Department of Mathematics and Computer Science, FSTS, Hassan 1st University Settat 26000, Morocco	Abdelouhaed Kriouile LAVETE Laboratory, Department of Mathematics and Computer Science, FSTS, Hassan 1st University Settat 26000, Morocco	Taoufiq Gadi LAVETE Laboratory, Department of Mathematics and Computer Science, FSTS, Hassan 1st University Settat 26000, Morocco
--	---	---	--	---

ABSTRACT

Currently, agile enterprises are seeking to identify, represent and document the Business rules (BRs) which become one of the most effective ways to express business knowledge. So due to their impact on the information system, detecting and identifying mistakes become an important topic of great concern for the development of information systems. In this paper, we provide a review of several methods treating mistakes in business rules, and present a criteria-based comparison. This work is an outline of future work to integrate business rules filtering in the MDA approach. Also the results of this study will make easier, to provide a relevant method for verification and resolution of the consistency of business rules have a significant impact on the MDA approach.

General Terms

Software engineering

Keywords

Business Rules, detection, resolution, structural errors, SBVR

1. INTRODUCTION

Business rules present an important element of enterprise architecture. They implement the enterprise goals and strategies, govern business process, and provide guideline to business applications - which are implemented within the information systems. Morgan defines a business rule in [8] as "a compact statement about an aspect of a business. The rule can be expressed in terms that can be directly related to the business, using simple, unambiguous language that's accessible to all interested parties: business owner, business analyst, technical architect, and so on".

Business rules present an important element of enterprise architecture. They implement the enterprise goals and strategies, govern business process, and provide guideline to business applications - which are implemented within the information systems. Morgan defines a business rule in [8] as "a compact statement about an aspect of a business. The rule can be expressed in terms that can be directly related to the business, using simple, unambiguous language that's accessible to all interested parties: business owner, business analyst, technical architect, and so on". Concerning the importance of BRs, there is always a possibility of mistakes in

rules when put into practice by experts engaged in different fields. Thus, identifying anomalies in business rules is a crucial step in development of the Information Systems (IS), which presents bad drawbacks in terms of losing of time and increase the cost of the maintenance. Therefore the verification and the resolution of the business rules play an important role to implement successfully the IS. The verification focuses on detecting structural errors which result from interaction between rules. Structural errors may influence the consistency of rule inference which involves redundancy, inconsistency, incompleteness, overlapping and circularity [10 -11].

Several methods are used for treating some of these structural errors, detecting and re-solving them, such as [7] which describes a theory and the resulting development environment for performing conflict resolution during development to exclude the inference engine for systems using propositional logic. In [12], the authors presents a novel rule model known as SOECAP (Subject, Object, Event, Condition, Action, Post-condition) treating the conflict resolution set dynamically using Vague set Theory. Then in [4] the authors provide an uncertain rule-based knowledge conflict treatment algorithm by integrating a set of decisions and an uncertain inference As well as in [6] presents a method for verifying the consistency of business rules using alloy model.

The contribution of this paper is a comparative study on relevant methods treating the structural errors of business rules and is used to assess them according to a set of comparative "creations". The aim is to determine whether these methods have the capability of re-solving the structural errors in business rules or not. This paper is organized as follows: in section 2, we give an overview of the methods found in a survey. The comparison, analysis and discussion are presented in section 3. Finally, the section 4 concludes the paper.

2. METHODS TREATING STRUCTURAL ERRORS OF BUSINESS RULES

We present in this section different methods used for treating anomalies in the business rules based on different strategies such as the meta-language, Petri nets, logic programing vague set theory, and no-inference engine. We have limited our study on the most known methods of each strategy. Some of

these methods present algorithms for checking anomalies and others present algorithms for resolution.

2.1 EVA METHOD

The EVA method [2] offers a wide field validation tools to check the anomalies of knowledge based system. It uses a declarative meta-language (higher-order language) to describe restrictions necessary to validate the redundancy, consistency, circularity, overlapping and correction of a knowledge base. This method was extended to non-monotonic logic [1] and it proposes three types of checking: structural, logical and semantic.

Therefore it offers the most comprehensive detection. It considers a rule in a knowledge base as a constraint or as a functional mapping from the left-hand side (LHS) of rule to the right hand side (RHS). To treat the redundancy and the overlapping it checks the duplication and subsumption rules, if the LHS of first rule subsumes or is the

same as the LHS of the second rule, then whenever one of the rules can be rejected. Therefore, any actions in the RHS of second rule which also occur in the RHS of first rule can be eliminated. If after the elimination no actions are left in the RHS of R2, then R2 can be eliminated. In order to check cycles, the method introduces the metapredicate, it defines the cycle rules using the term of basic predicates, or the terms of basic and derived predicates. For example, the predicate "ancestor" may be defined in terms of the basic predicate "parent". To find out logical inconsistencies, the method uses the meta-predicate "incompatible" as a goal for the logic checker. It initiates backward chaining to solve the goal. If it found answers for the goal, then the Knowledge Bases is inconsistent, else it is not inconsistent. However this method provides only the detection of knowledge based, it offers the most comprehensive detection. The fig. 1 presents the validation objectives and functionality of EVA.

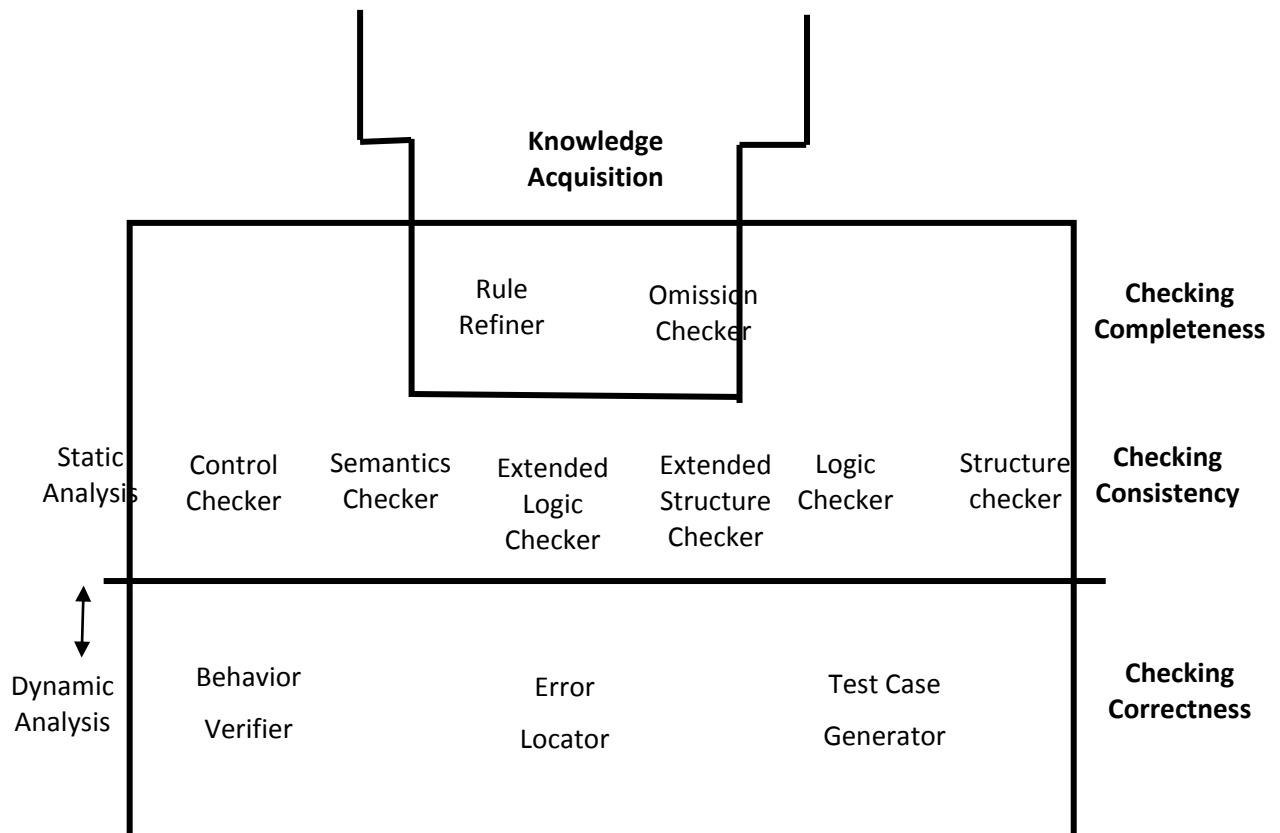


Fig 1: the validation objectives and functionality of EVA [2]

2.2 PREPARE method

PREPARE (PREdicate/transition nets and PAttern REcognition method) [5] uses predicate/transition nets to represent knowledge. These nets are special classes of Petri nets and verification is done through syntactic patterns of recognition. Inconsistent, redundant, subsumed, circular and incomplete rules in a knowledge base are then defined as patterns of the Predicate/Transition net model, and are detected through a syntactic pattern recognition method.

These nets are a graphic representation of predicate logic. PREPARE provides a approach to verify the correctness of knowledge bases. It checks for inconsistency, redundancy, circularity, and incompleteness in a knowledge base. By using

the component Transformer, it translates rules and facts of a knowledge base into a Pr/T net representation understandable by a computer; and in order to detect and locate the boundaries of the inconsistent, redundant, circular and incomplete patterns; it uses the Scanner component and stores the result in a table. After the detection of the anomalies the component Formulator encodes and formulates a string for each inconsistent/redundant/ circular pattern; and saved the results in a second table. After that the component Classifier determines the pattern type by recognizing the strings, produced by the Formulator, and discovers the incomplete type by analyzing the results obtained by the Scanner. Fig. 2 taken from [5] shows the relationships of these major components.

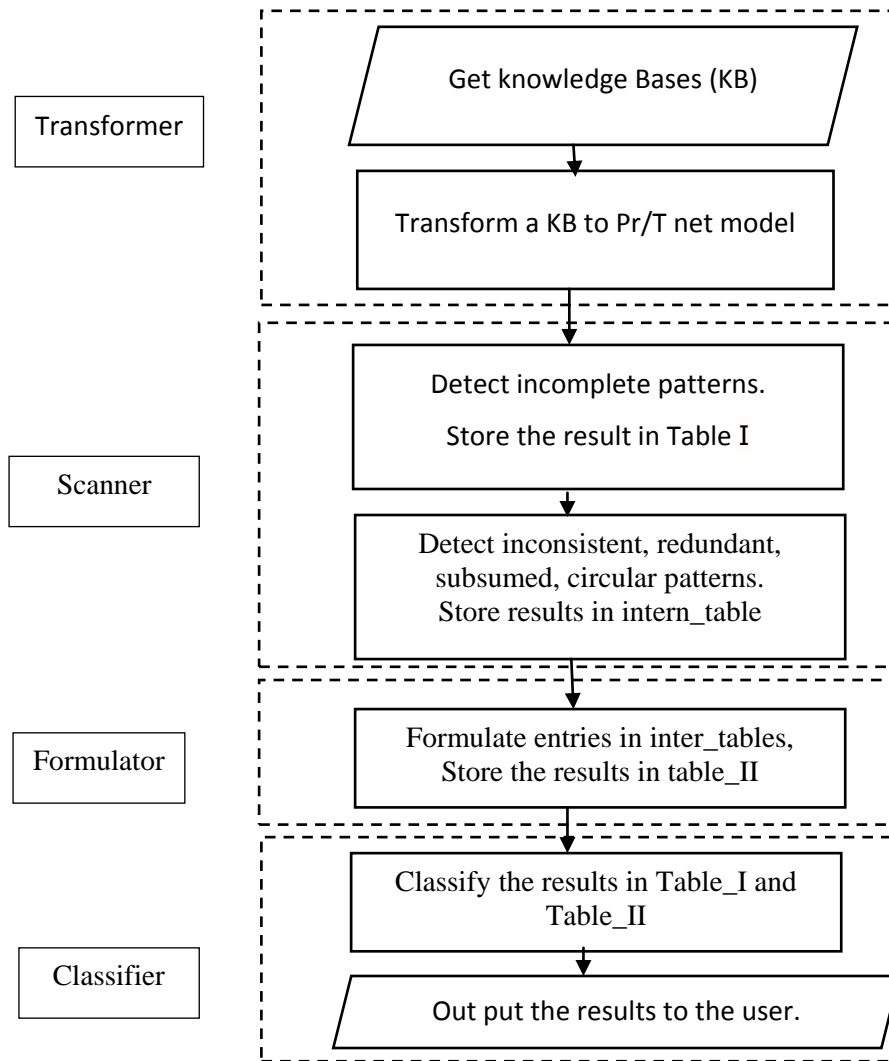


Fig 2: Relationships of major components of PREPARE method [5]

2.3 Method using Vague set

This method provides an algorithm to detect and resolve the conflict of BRs through Vague set Theory [12] which is introduced to measure the preference relationships among the conflict rules with the help of post conditions in SOECAP rules, and then a new conflict resolution method is put forward. The SOECAP model is an extension of the ECA (Event, Condition, Action) model with the concepts subject, object and post-condition. The subject and object are introduced to depict the trigger and effective object of the rule, and implies the logical

relations between the two types of entities involved in the rule. Post-Condition is introduced to present the system state constraint imposed by rule actions, aiming at the problems of detection and resolution on rule conflict. The algorithm using in this method based on vague set is presented as follow in fig. 3, it takes as input the set of rules represented by S, and the R represents the resolution set corresponding to S, A as the rule action set, and function get-Preference is used to compute the preference value from rule resolution set to conflict rule. R represented as a rule, r.o is the object of r, and r.p is the postcondition of r

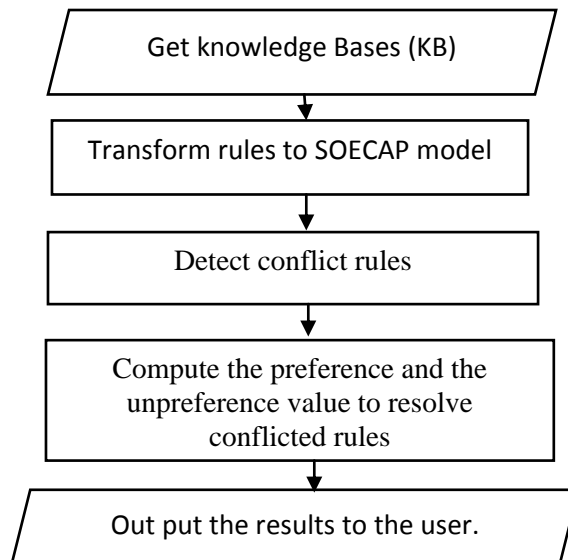


Fig 3: the algorithm using in this method based on vague set

2.4 URKCTA [4]

This method uses group decision, with an algorithm that has a confidence factor. In the algorithm, a “reliability factor” refers to the reliability level of the conflicting or redundant rules, while the “certainty factor” indicates the existence of the knowledge itself. A “certainty reliability index” is used to show both the existence of the knowledge itself and its reliability. For conflicting or redundant rules, it is suggested that the knowledge with a higher reliability factor be chosen. For uncertain rule-based knowledge, the study integrated group decision and uncertain inference concepts to put

forward URKCTA in which RF represents the reliability of the knowledge with conflicting or redundant rules; cf represents the certainty of the existence of the knowledge itself; and CRI comprehensively indicates the existence of the knowledge itself and its reliability. The study revealed the CRI for conflicting or redundant rules, and comprehensively showed the existence and reliability of the rule itself for use in determining the certainty and reliability of the cited knowledge. This method is suitable in situations with BRs conflicts that cannot be resolved. Fig. 4 taken from [4] represents the URKCTA architecture.

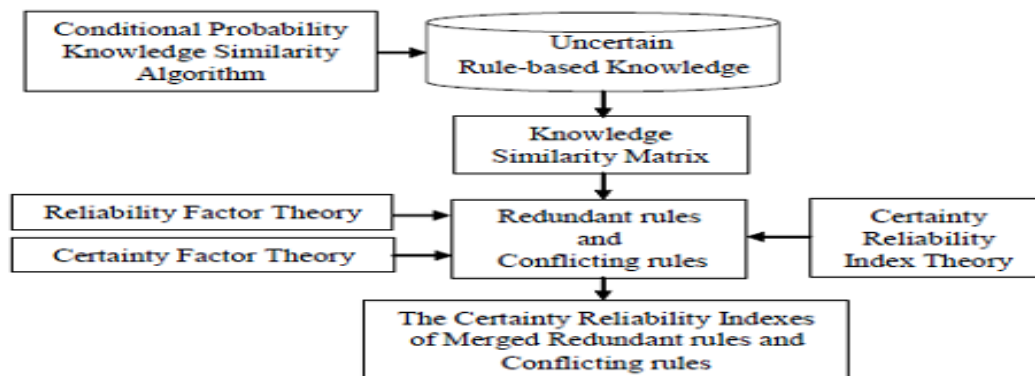


Figure 4: The architecture of URKCTA [4]

2.5 Method Using LogicProgramming

This method [3] uses a declarative policy description language PDL, in which policies are formulated as a sets of ECA (Event, Condition, Action) rules and it presents a framework for detecting action conflicts and finding resolutions to these conflicts. Conflicts are captured as violations of action constraints. The semantics of rules and conflict detection and resolution are defined axiomatically using logic programs. Given a policy and a set of action constraints, the framework defines a range of monitors that filter the output of the policy to satisfy the constraints. This method uses predicate symbols (occ, exec, block, ignore and accept) to capture the conflict resolution rules. occ is the input epoch of a monitor, the exec

is the output of the policy, and accept is the output of a monitor.

2.6 Method Using Alloy [6]

This method gives a tool for the evaluation of the consistency of a set of BRs, and proposes an incremental form of process evaluation of a set of BRs, that this set is updated in a continuous way through the life cycle of BRs applications. This method verifies the overlapping, the redundancy, and the conflicts of BRs. After the representation of rules by sbvr it uses the model alloy which is a language to describe the structures as the tool to explore them. The process of The method proposed to verify the coherence of a set of BRs composes of five stages, to see the figure 5. First stage of the method consists in identifying the BRs and specifies them in a

structured way, by using the business language for the representation. Secondly, it builds a conceptual model, describing the business concepts and their relations. Thirdly, it transforms rules in a alloy Model, to build coherent set of BRS. Fourthly, Using a rule at a time, present the new BR in the producing model, possible inconsistency situations. When this occurs, update the whole of BRs.

And finally, it repeats the previous stage, until no inconsistencies in BRs are found and the whole of retunes of to coherent situation. Therefore, the power of this method consists with its capacity to find the majority of the anomalies of the business rules existed like the redundancy, Overlapping and the conflict.

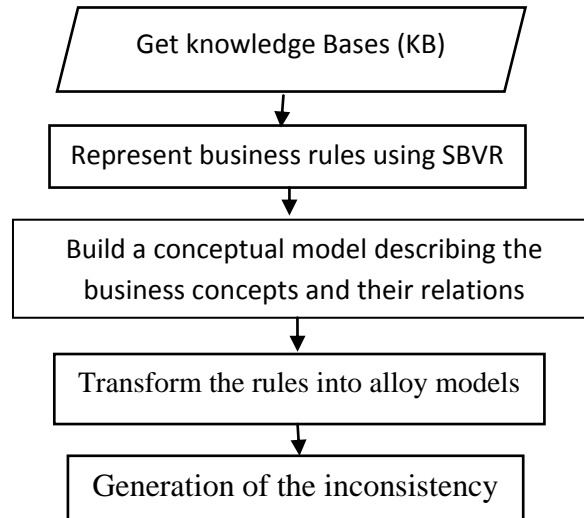


Figure 5: Alloy steps for verifying the consistency of BRs

2.7 NIET method [9]

This method proposes No Inference Engine Theory (NIET) for rule-based systems. It performs conflict resolution using propositional logic during development to eliminate the inference engine for systems. It uses verification criteria and solution strategies and derives four classes of rules and their rule ordering strategies. Rule based systems using FRS (First Rule Satisfied) rules contain four classes of rules. The Deterministic contains rules that satisfy every combination of condition values that exist. The incomplete class gathers incomplete rules meet the criteria of Consistency, but are

incomplete because not all combinations of condition values will satisfy a rule. The exception class has more specific rules which are the most desirable solutions because they represent exceptions to more general rules. And at last the belief class has the most desirable solution is the one with the highest confidence. Consequently the use of a prototype IDE that use the class of rule to implement the appropriate conflict resolution strategy during development eliminate the need for run-time conflict resolution and therefore eliminating the need for a physical inference engine. The fig. 6 shows the steps using by NIET method.

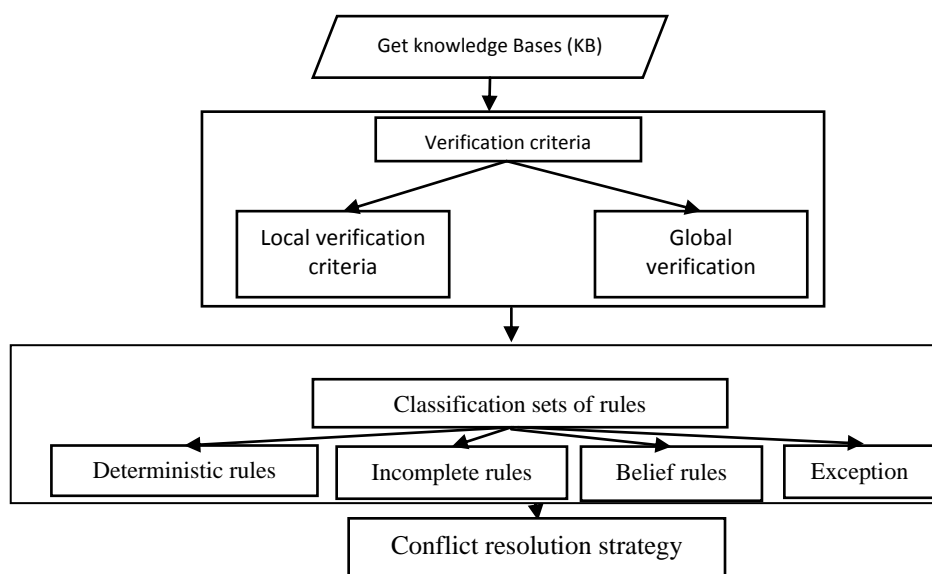


Figure 6. Shows the steps using by Niet method.

3. COMPARISON, ANALYSIS AND DISCUSSION

In this section we will start by presenting comparative criteria, and thereafter we compare the proposed methods on the basis of these criteria:

3.1 Comparative criteria

The typical inconsistencies in business rules specification are: redundancy, overlapping, conflict, incompleteness and circularity [5]. Redundancy: occurs when two rules have the same set of conditions (with conditions possibly arranged in a different order) and the same conclusion. Then, one of them is said to be redundant [5]. Overlapping: occurs when one rule

is wholly or partly contained within another [8] or when two rules have the same conclusion and one rule has a more restrictive condition. The more restrictive rule can be subsumed by the less restrictive rule on the grounds that whenever the former succeeds, the latter also succeeds [5]. Conflict: occurs when two or more rules produce contradictory results [8] or their conditional parts consist of the same set, but the conclusions are mutually exclusive [5]. Incompleteness: occurs when all the information necessary to produce a conclusion does not exist. This may be caused by gaps left inadvertently, uncertain knowledge or lost track of the grown knowledge base [5]. Circularity: Given a set of rules, if they are defined such that one rule references another in a circular fashion, and then they are called circular rules [5]. It occurs when there is no end to the rule inference [4] and the rules that lead to an infinite loop.

3.2 Analysis and discussion

The result from the analysis of comparative criteria is presented in table 1. The rows in the table are five criteria, the columns are seven; each column presents a method studied beforehand (Section II). As we can see from table 1, among

the seven methods, four can detect the redundant rules and one of them resolves redundancy

Regarding incompleteness, only two methods can detect it. Concerning circularity only the two methods detect it and any one resolves it. Mean way all methods detect conflict but only four resolve it. Finally one in two methods resolves overlapping and both of them cover the problem. From what precedes, we see that no method perfectly meets the five criteria for inconsistent business rules. With regard to the results of the comparative criteria for inconsistent business rules, only one studied method resolves redundancy and conflict which is URKCTA method, whereas base on vague set method, logic programming method and NIET method resolve the problem of conflict between BRs. Methods PREPARE method and EVA method detect redundancy, circularity and conflict between BRs. Besides PREPARE method is distinguished by its capacity to detect incompleteness and URKCTA method is distinguished by its capacity to resolve overlapping. The remaining methods satisfy one or two criteria.

4. CONCLUSION

The relevance of the business rules in the Information System (IS) of the enterprise makes the treatment of business rules an important topic to reduce the problems associated with the IS, and to lower the cost of developing and maintaining computer applications. Our work tries to help researchers achieve a proper understanding of the state of the art and indicates directions for future research to fill the gap felt in this context, with a comparison based on the criteria of existing works for this stage of detection and resolution of inconsistency rules. As a future work, It would be trying to fill the gaps in methods of treating BRs anomalies, and, thus, propose a method which ideally meets the evaluation criteria proposed in our study. This method must be able to detect and resolve redundancy, circularity, incompleteness and conflict of BR

Table 1. Comparison summary of the methods proposed in this study

criterion name		EVA method	PREPARE method	base on vague set method	URKCTA method	Alloy method	logic programming method	NIET method
redundancy	Detection	Y	Y	N	N	Y	N	N
	Resolution	N	N	N	Y	N	N	N
incompleteness	Detection	N	Y	N	N	N	N	Y
	Resolution	N	N	N	N	N	N	Y
Circularity	Detection	Y	Y	N	N	N	N	N
	Resolution	N	N	N	N	N	N	N
Conflict	Detection	Y	Y	N	N	Y	Y	Y
	Resolution	N	N	Y	Y	N	Y	Y
overlapping	Detection	Y	N	N	N	Y	N	N
	Resolution	N	N	N	Y	N	N	N

5. REFERENCES

[1] B. Gerhard, D. Jürgen et K. Kurt, Nonmonotonic Reasoning - An Overview, Stanford: CSLI publications, 1997.

[2] C.L. Chang, J.B. Combs, R.A. Stachowitz, "A Report on the Expert Systems Validation Associate (EVA)" Expert Systems With Applications, pp. 217-230, 1990.

- [3] C. Jan, "Conflict Resolution Using Logic Programming," *IEEE Transactions On Knowledge And Data Engineering*, vol. 15, pp. 244 -249, 2003.
- [4] C. MIN-YUAN et H. CHIN-JUNG, "A Novel Approach for Treating Uncertain Rule-based Knowledge Conflicts" *J. OF INFORMATION SCIENCE AND ENGINEERING*, vol. 25, pp. 649-663, 2008.
- [5] D. Zhang. et N. Doan., "PREPARE: A Tool for Knowledge Base Verification" *IEEE Transactions on Knowledge and Data Engineering*, pp. 983-989, 1994.
- [6] G. Denilson dos Santos, S. Eber Assis et J. A. Antônio, "A Method for Verifying the Consistency of Business Rules Using Alloy", *International Conference on Software Engineering & Knowledge Engineering*, Vancouver, Canada, 2014.
- [7] L. An et N. Wilfred, "Vague Sets or Intuitionistic Fuzzy Sets for Handling Vague Data: Which One Is Better?" *Lecture Notes in Computer Science*, vol. 3716 , pp. 401-416, 2005.
- [8] M. Tony, "Business Rules and Information Systems: Aligning It with Business Goals" Boston, MA: Addison-Wesley, 2002.
- [9] Richard, C. Hicks, "The no inference engine theory — Performing conflict resolution during development", *Decision Support Systems*, vol. 43, n° 12, p. 435-444, 2006
- [10] S. Motoi, S. A. Carlisle and S. Edward , "An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System" *AI Magazine*, vol. 3, pp. 16-21, 1982.
- [11] T. Wei-Tek, V. Rama et Z. Du, "Verification and validation of Knowledge-Based Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, pp. 202-212, 1999.
- [12] 12 Z. Qingchuan, Z. Guangping, X. Chaoen et Y. Yang, "A rule conflict resolution method based on Vague set," *Soft Computing* , vol. 18, n° 13, pp. 549-555 , 2013.