Experiences with Modeling and Verification of Regulations

Jan Vanthienen[1], Christophe Mues[2], and Stijn Goedertier[1]

[1] Department of Decision Sciences & Information Management,
Katholieke Universiteit Leuven, Belgium
Naamsestraat 69, B-3000 Leuven, Belgium
{jan.vanthienen;stijn.goedertier}@econ.kuleuven.be

[2] University of Southampton, School of Management, United Kingdom
Highfield Southampton, SO17 1BJ, United Kingdom
c.mues@soton.ac.uk

Abstract. Information system models commonly describe organizations in terms of the structure of the data they use, the organization of the processes they perform and the operations that will be executed. Yet to date, the models tend to neglect the rules and constraints under which the business operates. Frequently these are not represented or even documented, but hard-coded or hidden into implementations. Compliance with regulations and verification of business rules is difficult when a considerable amount of business knowledge is only represented in code or database constructions and is not modeled in an explicit way that is easy to understand and verify. When properly defined, decision tables offer a solution to several issues in the modeling and verification of regulations, such as guaranteeing consistency, correctness, non-redundancy and completeness.

1 Introduction: The Business Rules Approach

Building flexible and compliant systems does not work well without business rules and the ability to create and manage business rules in a well organized setting. Both practitioners and academics advocate to bring business rules out of obscurity, out of implementation and into the business management side [1] [2] [3]. The essence of the business rules approach is to describe, manage, verify, validate and automate aspects of the business function in a declarative instead of a procedural way . Because rules and regulations apply across processes, systems and applications, they should be modeled and managed separately.

Business rules and constraints have become a common topic in systems development and enterprise modeling. Unfortunately, a comprehensive, industry-wide approach to modeling and implementing business rules is lacking. Moreover, discussions on this topic frequently give little recognition to the amount of work that has been done in knowledge-based systems (KBS) technology in such areas as knowledge acquisition, validation and verification (V&V), inferencing, representation, etc.

Odell defines business rules as “declarations of policy or conditions that must be satisfied” [4]. The author suggests a taxonomy of business rules that distinguishes **constraints** and **derivation rules**. Constraints initiate, enable, govern, and limit the behavior of objects and agents to accomplish the goals or purposes of a system. Odell distinguishes three kinds of constraints: simulus/response rules, constraints on operations and constraints on data. Derivation rules derive a fact about an object, and are classified as inference rules or computations.

This taxonomy maps well to standard information concepts, including OO concepts. For example, operation constraints are pre- and post-conditions on object operations, and structure constraints are class invariants. However, stimulus/response rules, which state that when something occurs, something should be done, are not commonly supported by OO languages and are typically implemented outside of an OO application as triggers in the database. On the other hand, knowledge-based languages provide software support for stimulus/response rules using demons. Object-oriented techniques often ignore inferencing and reduce inference rules to procedural code, thereby violating the declarativeness of business rules. In order to separate code and data from rules, and add intelligence and flexibility to systems development, rule engines and rule management systems will have to be added.

2 Verification and Validation of Rules

Research in the verification and validation (V&V) community focuses on rule anomalies. A classification which is commonly used in the V&V community of knowledge-based systems can be found in Preece and Shinghal [5]. It considers the anomalies represented in Fig. 1.

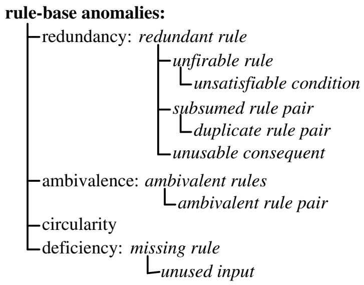


Fig. 1. A classification of rule base anomalies

In order to detect these anomalies, many tools have been developed. An important problem of automatic rule base checking is that these tools often face combinatorial explosion as the rule base increases. Therefore, it may be necessary to split the rule base or use heuristics to circumvent this problem.

Although anomaly detection tools can be effective, it would be wrong to rely completely on these tools to build high-quality systems. These tools are merely an add-on to the development environment, and mostly they possess no facilities to prevent anomalies during the design process.

3 Modeling Business Knowledge using Tables

A decision table is a tabular representation used to describe and analyze procedural decision situations, where the state of a number of conditions determines the execution of a set of actions. Not just any representation, however, but one in which all distinct situations are shown as columns in a table, such that every possible case is included in one and only one column (completeness and exclusivity).

Many variations of the decision table concept exist which look similar at first sight [2]. The most important criterion when distinguishing tables, is the question whether all columns are mutually exclusive (single hit versus multiple hit). If the columns are not exclusive, some combination of conditions matches more than one column, which may lead to ambiguity or inconsistency. As a result the overview over the columns is lost, and with it, the simplicity of inspection. For these reasons we do not consider these tables to be real decision tables, even though they are frequently mentioned in practice.

Different experiments have been conducted in order to compare the effectiveness of the decision table as a conditional logic representation tool with other formalisms. Details on these experiments can for example be found in Vessey and Weber [6].

Though originally used as a software engineering technique, decision tables have proven a useful aid in modeling complex decision situations of various sorts (see also Parnas [7], Heitmeyer [8]). In the field of business rules and regulations, there has been a renewed interest in decision tables over the past years.

Decision tables act as an *intermediate between the specification of a decision process and the real decision making act*, allowing overview and verification throughout the life cycle. A decision table is **condition oriented** and therefore effective in representing procedural knowledge, with such advantages as: overview, readability, consistency, completeness and correctness.

This structured enumeration of decision columns, however, is not the way in which procedural knowledge is acquired or specified. Most texts, procedures, laws, etc. are described in **action oriented**, partial or modular decision specifications, not suited for fast and correct decision making or easy verification. To this end, the decision table construction process allows to transform the action oriented *specification* into a condition oriented *representation*. The decision table being a representation mechanism, *optimization* is not the main concern. But optimization may become a concern, once the table has to be converted into an operational system or a manual decision making procedure. At this point, for example it may become important to minimize the total test time or the number of questions. The advantage of the decision table approach is that implementation aspects can be separated from representation aspects through transformation. The decision table approach, therefore, unifies these three complementary aspects of a decision situation, as depicted in Fig. 2.

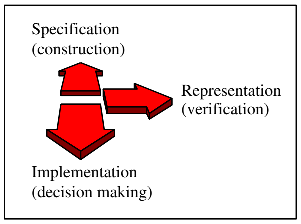


Fig. 2. Three aspects of a decision situation

The role of the decision table formalism therefore is not limited to the occasional and isolated use. But it covers the full trajectory of the development life cycle: starting from the early stages of knowledge acquisition and structuring, up to and including the final transformation of decision tables into existing knowledge base tools and products. These decision tables are obtained through the combined use of techniques from knowledge acquisition and discovery. V&V activities are seen as an integral part of this modeling process. At a final stage, the resulting body of decision tables can serve as a basis for implementation.

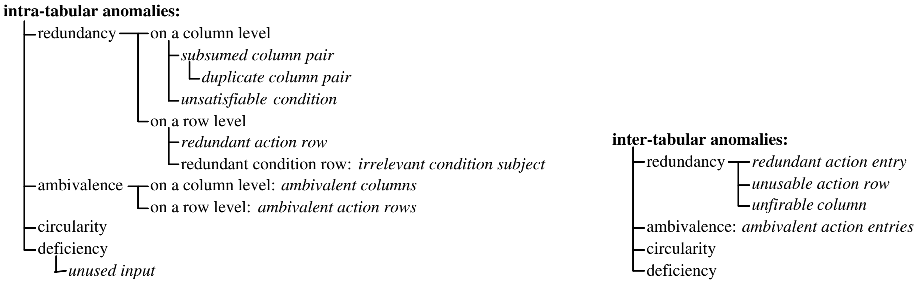
4 Decision Tables for Verification

The problems of validation and verification have led to the occasional use of schemes, tables or similar techniques in knowledge representation and validation. As has been reported earlier in Cragun and Steudel [9], most of these verification and validation problems can be solved by the decision table technique. But anomaly detection systems based on decision tables or related formalisms, mainly suffer from the drawback that they fail to find anomalies in the rule base which occur over inference chains. Most of those systems can only detect anomalies between simple pairs of rules. Also most of these systems do not check for circularity.

In order to overcome these limitations, a different perspective in developing the anomaly detection facilities has been taken. Most of these limitations do not hold anymore if the decision tables are properly defined. We advocate the use of decision tables as a modeling technique on its own, and not only as a means towards verification of regulation systems, such that the anomaly detection component is integrated in the modeling phase of the system.

Starting from the classification by Preece it was investigated how verification anomalies can be detected in a decision table based system [10]. A distinction is made between tabular anomalies and inter-tabular anomalies. The former occur in a single table whereas the latter originate from interactions between several tables. The classification given in Fig. 3(a) is used to describe intra-tabular anomalies. As only single hit tables are allowed, where every possible combination of condition values is represented in one (completeness) and only one column (exclusivity), most of the traditional anomalies cannot even occur or are easy to detect.

To check a system of interrelated tables for anomalies, it is not sufficient to check each of the components separately. It also has to be checked whether there exist some inter-tabular anomalies. These anomalies can only be detected by inspecting the system as a whole, because these anomalies result from the interactions between two or more tables. A classification of inter-tabular anomalies is depicted in the Fig. 3(b). Most of the limitations shown by anomaly checking systems based on decision tables can be solved. These include the possibility to check for anomalies over chains of rules and the check for circularity [11].



(a) Intra-tabular anomalies

(b) Inter-tabular anomalies

Fig. 3. A classification of tabular anomalies

5 Experiences with regulations

The use of decision tables need not be restricted to visualization or verification of existing regulations. The technique is well applicable throughout the lifecycle, from design to maintenance.

- **Designing regulations.** When designing new regulations, the decision table technique can be used to to indicate which combination of criteria should lead to which outcomes, thereby guaranteeing combinatorial completeness and consistency. The ability to focus on well-defined situations (columns) has proven a to be a valuable communication advantage.
- **Checking the quality of the regulations.** When a written regulation exists, the decision table might be constructed in order to visualize the specifics of the regulation. Transforming the text into the decision table representation for validation and verification purposes, has traditionally been a major application area of decision tables.

- **Impact analysis.** Decision tables easily allow to examine the impact of changes in one part of a regulation on the total result. The net effect of changing conclusions for some combinations of condition values is immediately visualized.
- **Generating test cases.** Because all possible situations are represented, the decision table is an ideal source to generate test cases. Instead of random testing, attention can be paid to the full coverage of a range of criteria, boundary value analysis, etc.
- **Restructuring the regulation.** The decision table can not only be used to verify or implement the regulation. Once the table is finished, it can be used to redesign or reorder the text, refine and simplify the description or change the fundamental structure based on the decision table representation. Regulations are often described in the order in which they are conceived, but that is not necessarily the best order for execution. Therefore, if a new description of the regulation is desired, it can be derived from the decision table in a more orderly fashion by examining the application field of each of the actions and describing it in its most compact logical form.
- **Applying the regulations.** Once completed, the decision table can be used to generate code or rules, or can be used as an execution mechanism. If the implementation is straightforward, maintenance of the regulation can be performed directly on the decision table.

We have used decision tables in a large number of applications and environments. Some examples of the more common areas of experience include :

- verification and visualization of laws and regulations [12],
- modeling and verification of complex managerial decision situations in general [13],
- comprehensible representation of scoring models in banks and insurance companies [14]

The use of decision tables in regulations is e.g. illustrated in the Handipak system [12], a knowledge based system with regard to financial benefits for the disabled in Belgium. Its main goal was to support first line social workers when helping the disabled with the introduction of their applications for benefits.

The knowledge base of the Handipak advisory system was developed by using decision tables. The relevant law was couched in 45 mutually related decision tables. The relations between the different tables were specified in a number of control tables. The regulation with regard to financial benefits for the disabled was divided into subunits, that respectively concern: the validity of the application; the applicability with regard to nationality, age, and residence; the granting conditions with regard to category or other social benefits; the calculation of the maximal benefit and the revenues to be deduced; the modalities of the payment. For each of the regulatory subunits, one or more mutually related decision tables were developed.

The analysis of the proposed regulation by means of the decision table technique enabled the authors to eliminate a considerable number of ambiguities in the bill before it was published.

6 Conclusion

Over the years there has been a major change in research and application areas of decision tables. Current decision table applications mainly benefit from the representational and verification advantages of the technique. The key concepts fit very well with recent developments in the areas of business rules. Based on numerous experiences with decision tables, a refined decision table concept has been proposed, which supports this vision.

References

1. Ross, R.G.: Business Rule Concepts - Getting to the Point of Knowledge. Business Rules Solutions, LLC; 2nd edition (2005)
2. Vanthienen, J.: 50 ways to represent your rule sets. In Business Rules Journal **7**(1) (2006)
3. Goedertier, S., Vanthienen, J.: Rule-based business process modeling and execution. In: Proceedings of the IEEE EDOC Workshop on Vocabularies Ontologies and Rules for The Enterprise (VORTE 2005). CTIT Workshop Proceeding Series (ISSN 0929-0672), Enschede (2005)
4. Odell, J.: Business rules. Object Magazine (1995) 53–56
5. Preece, A.D., Shinghal, R.: Foundation and application of knowledge base verification. International Journal of Intelligent Systems **9**(8) (1994) 683–701
6. Vessey, I., Weber, R.: Structured tools and conditional logic: An empirical investigation. Commun. ACM **29**(1) (1986) 48–57
7. Parnas, D.L.: The tabular method for relational documentation. Electr. Notes Theor. Comput. Sci. (**44**(3)) 1–26
8. Heitmeyer, C.L., Jeffords, R.D., Labaw, B.G.: Automated consistency checking of requirements specifications. ACM Trans. Softw. Eng. Methodol. **5**(3) (1996) 231–261
9. Cragun, B.J., Steudel, H.J.: A decision-table-used processor for checking completeness and consistency in rule-based expert systems. Int. J. Man-Mach. Stud. **26**(5) (1987) 633–648
10. Vanthienen, J., Mues, C., Wets, G., Delaere, K.: A tool-supported approach to inter-tabular verification. Expert Systems with Applications **15** (1998) 277–258
11. Mues, C., Vanthienen, J.: Efficient rule base verification using binary decision diagrams. In Galindo, F., Takizawa, M., Traunmüller, R., eds.: DEXA. Volume 3180 of Lecture Notes in Computer Science., Springer (2004) 445–454
12. Vanthienen, J., Robben, F.: Developing legal knowledge based systems using decision tables. In: Proc. of The Fourth Int. Conf. on A.I. and Law, June 15-18, Amsterdam,. (1993) 282–291
13. Vanthienen, J., Dries, E.: Illustration of a decision table tool for specifying and implementing knowledge based systems. In: Proc. of The Fifth Int. Conf. on Tools with Artificial Intelligence (TAI), November 8-11, Boston (Mass.). (1993) 198–205
14. Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit-risk evaluation. Manage. Sci. **49**(3) (2003) 312–329