

1-1-2011

# A Structured Analysis of Business Rules Representation Languages: Defining a Normalisation Form

Martijn Zoet

*HU University of Applied Science*, [martijn.zoet@hu.nl](mailto:martijn.zoet@hu.nl)

Pascal Ravesteyn

*University of Applied Science Utrecht*, [pascal.ravesteijn@hu.nl](mailto:pascal.ravesteijn@hu.nl)

Johan Versendaal

*Utrecht University*, [johan.versendaal@uu.nl](mailto:johan.versendaal@uu.nl)

Follow this and additional works at: <http://aiselaisnet.org/acis2011>

---

## Recommended Citation

Zoet, Martijn; Ravesteyn, Pascal; and Versendaal, Johan, "A Structured Analysis of Business Rules Representation Languages: Defining a Normalisation Form" (2011). ACIS 2011 Proceedings. Paper 20.  
<http://aiselaisnet.org/acis2011/20>

## A Structured Analysis of Business Rules Representation Languages: Defining a Normalisation Form

Martijn Zoet  
HU University of Applied Sciences Utrecht  
Utrecht, The Netherlands  
Email: martijn.zoet@hu.nl

Pascal Ravesteyn  
HU University of Applied Sciences Utrecht  
Utrecht, The Netherlands  
Email: pascal.ravesteijn@hu.nl

Johan Versendaal  
Utrecht University  
Utrecht, The Netherlands  
Email: johan.versendaal@uu.nl

### Abstract

*Business rules play a critical role during decision making when executing business processes. Existing modelling techniques for business rules offer modellers guidelines on how to create models that are consistent, complete and syntactically correct. However, modelling guidelines that address manageability in terms of anomalies such as insertion, update and deletion are not widely available. This paper presents a normalisation procedure that provides guidelines for managing and organising business rules. The procedure is evaluated by means of an experiment based on existing case study material. Results show that the procedure is useful for minimising insertion and deletion anomalies.*

### Keywords

Business Rules, Business Rules Management, Normalisation, BRM.

### INTRODUCTION

Business process management and business rules management both study the management and execution of tasks (Van der Aalst et al. 2003). However, both do so from different perspectives. Business process management (BPM) takes an activity/resources viewpoint while business rules management (BRM) approaches tasks from a guideline/knowledge viewpoint. Integrating the two viewpoints has been of interest to scientist as well as practitioners (Gottesdiener 1997, Zoet et al. 2011). Of special interest are analytical tasks that determine a decision for a specific case based on domain-specific business rules. The reason for this is that a direct relation between the two management practices can be established. On the one hand such activities are modelled and executed within business processes while on the other hand they need transactional sequencing business rules for guidance, such that consistent decisions can be made (Zoet et al. 2011). Examples of such tasks are “determine policy renewal method”, “determine candidate ranking” or “determine risk level of applicant”.

Business process modelling techniques have originally not been intended to model rule component. Yet, currently a wave of BPM-systems is being released that offer both process and rules modelling techniques (Dominguez 2009,Cordys 2010,Pegasystems 2011). As more options to integrate become available and the usage of business rules modelling techniques within BPM-systems increases, manageability of rules supporting business processes becomes an important issue. Also, to remain competitive, organisations are increasingly urged to adapt to changes in their business environment, representing another force that will raise manageability questions. However, scientific research with respect to business rules modelling guidelines that address manageability in terms of anomalies such as insertion, updates and deletion is scarce (Van Thienen and Snoeck 1993).

This paper extends understanding of business rules modelling guidelines by addressing manageability in terms of insertion, update and deletion anomalies based on the following premises. Similar to previous research, we consider relational theory as the foundation for our guidelines. Dissimilar to previous research, we do not focus on one specific language/visual syntax (Van Thienen and Snoeck 1993) but we start analysing mainstream decision rules modelling languages and build our approach from this. We posit that a preferred form of

structuring business rules could comprehend most common rules languages. With these premises, the specific research question addressed is: “*How can transactional sequencing business rules guided analytical tasks be normalised such that optimal manageability is realised?*” Answering this question will help practitioners better manage business rules that support analytical activities in business processes.

The remainder of this paper proceeds as follows. The next section provides a context by describing analytical tasks, business rules and relational theory. The third section describes the construction of the actual normalisation procedure. Section four presents the results of an experiment based on case study data. The final section summarises the study’s core findings, contributions as well as its limitations.

## THEORETICAL FOUNDATIONS

The purpose of a decision or analytical task is to determine a conclusion for a specific case based on domain-specific norms. In general the process of deriving a conclusion from specified norms can be described as follows (Breuker and Van de Velde 1994). First, data specific to the case at hand is collected by executing previous tasks or by consulting documents, software or other resources. This data is compared with predefined norms (transactional sequencing business rules) that are applicable to the case. This comparison leads to a specific value that in turn contributes to formulating the decision. Consider a policy renewal process at an insurance firm in which the task “determine risk level of applicant” is executed. First data is collected from and about the applicant. Secondly this data is compared with predefined norms defined by the insurance organisation. After this comparison a decision is made whether to insure the applicant and at what rate. The example and definition above demonstrate why analytical tasks are at the intersection of the BPM and BRM domain, see also figure 1. On the one hand activities need to be executed and coordinated to collect data and assemble information. This being the focus of BPM, which uses methods, techniques and software to design, enact, control and analyse operational processes (Van der Aalst et al. 2003). While on the other hand specific tasks, in this case the determination of the risk levels, are guided by transactional sequencing business rules. The use of methods, techniques and software to design, enact, control and analyse business rules is the focus of BRM (Zur Muehlen and Indulska 2009).

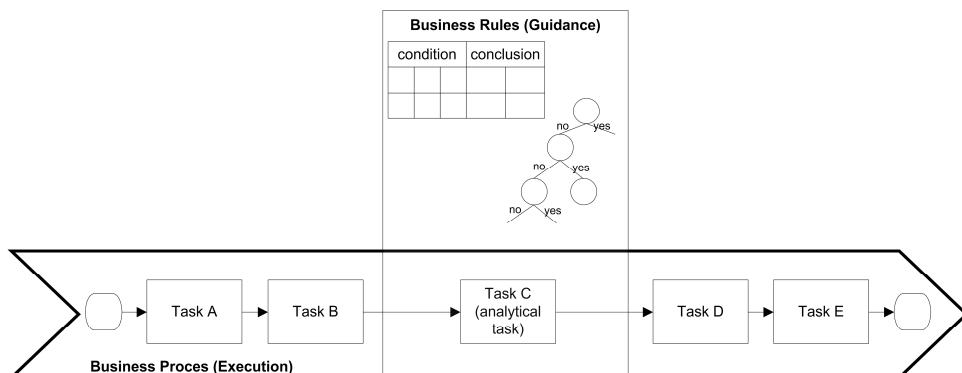


Figure 1: Intersection of business process and business rules

Business rules management research as a discipline/sub-field is relatively young. Yet, the BRM sub-field can draw from research on expert systems and knowledge management which both have a rich history. Unfortunately, research executed within the field of expert systems has lost its connection with industry some time ago (Arnott and Pervan 2005). In an elaborated survey Arnott and Pervan (2005) identified this problem and list the following reasons for its existence: almost no theory refinement research is executed, poor identification of clients and users, almost no actual case studies are executed and -maybe the most important reason- research is simply focussing on the wrong application areas. One upcoming application area where there are identifiable clients and users are business rules used in the context of the analytical tasks: transactional sequencing business rules. To the best of our knowledge no research addressing the manageability in terms of anomalies such as insertion, update and deletion of business rules has been conducted in the field of expert systems. Some research regarding this subject can be identified in the knowledge management community (e.g. Van Thienen and Snoeck 1993). Van Thienen and Snoeck (1993) report that maintainability is an issue with regards to decision tables and knowledge management systems and little research has been conducted to address this issue. In their study decision tables are used to represent business rules. Based on relational theory and database normalisation they propose guidelines to factor knowledge thereby improving maintainability. However, instead of formulating one common procedure they also propose multiple exceptions to the normal forms. These exceptions have to be formulated because of the foundation of their research: decision tables. We

are in agreement with Van Thienen and Snoeck (1993) that a procedure addressing anomalies should be created and that relational theory is a proper foundation. But we argue that a broader view including more business rules modelling languages should serve as a foundation to formulate such a procedure.

The definition of the term relational used in this paper is adopted from the mathematical domain, more specifically relational algebra theory (Codd 1970). Relation theory has received much attention during the last four decades, since popularised by Codd (1970) for database normalisation. It states that a relationship exists ( $R$ ) on a given set ( $S_1, S_2, S_n$ ) if it is a set of n-tuples from which the first element is of  $S_1$  its second element is of  $S_2$ , and so on (Codd 1970). Most authors (Codd 1970, Kent 1984) represent such sets by means of two-dimensional arrays. With this we have indentified the problem we want to solve as well as the theoretical foundation for the design of the artefact. The evaluation of information system artefacts can be conducted on various elements such as functionality, completeness and performance (Hevner et al. 2004). An error common in design research is to start without clear goals in mind (Hevner and Chatterjee 2010). And even when goals are set they can be unsystematic, use incorrect measures and invalid evaluation techniques. To overcome such problems the underlying hypotheses as well as their measurements must be clear. According to this reasoning we propose the following hypotheses:

Hypothesis 1: current decision / transactional sequencing business rules modelling languages can be translated to a unified view by means of relation theory and applied to analytical tasks

Hypothesis 2: normalisation of transactional sequencing business rules has a positive effect on the average number of tuples affected by anomalies.

## **DEFINING A NORMALISATION PROCEDURE FOR BUSINESS RULES**

We consider existing decision business rules modelling languages as the foundation of our normalisation procedure. Accordingly, before defining a normalisation procedure first the fit between existing languages and relational theory has to be established. Establishing this fit can be broken down into three steps. First a choice has to be made which modelling languages to select for comparison. Secondly, analysis has to be conducted regarding the difference and synergy of the rules modelling languages. When a high synergy between modelling language exist it is likely that a common format for relational theory can be found. Lastly, the actual format for the relation theory has to be defined.

Since a relatively high number of business rules modelling languages exist within scientific as well as professional literature a decision, for practical reasons, has to be made which of these languages to select for our analysis. The languages chosen have the following characteristics: they are well-known/common within the field, and they have served as basis for most vendor specific and scientific languages. We consider these requirements fulfilled when (1) a language is mentioned in at least five different books randomly selected from a list of books addressing business rules (management) and (2) a language is considered as an artefact for addressing decision-making issues within scientific research. During the first step ten books (Buchanan and Shortliffe 1984, Morgan 2002, Von Halle 2001, Ross 2003, Chisholm 2004, Graham 2006, Ligêza 2006, Schacher and Grässle 2006, Browne 2009, Ross 2009, Boyer and Mili 2011) were randomly selected and searched for business rules modelling languages. During the second step all modelling languages have been searched in scientific research databases to indentify if the selected techniques have been applied in cases for decision making. This resulted in the following list of six languages: if-then rules (Rivest 1987), decision tables (Kohavi 1995), decision trees (Quilan 1986), score cards (Morrow et al. 2000), event, condition & action rules (Dayal 1988) and event condition action alternative rules (Heimrich and Specht 2003) which are mapped to relational theory.

A technique used to indentify differences and overlap between concepts or constructs in ontology's, languages and visual syntax is representational difference analysis (Hubank and Schatz 1994, Zur Muehlen and Induslka 2010). Representational difference analysis originates from medical and biological research but has since been adopted by multiple fields including information systems research (Green and Rosemann2004). The representational difference analysis of the six business rules languages is summarised in table 1. Each intersection between languages contains two cells indicating the conceptual and relational differences. Conceptual difference indicates the number of non-overlapping constructs between the two languages. The difference in existing relations and plurality between concepts that are present in the two languages is displayed in the second cell.

The analysis shows two clusters of languages that display high similarities. Decision tables, decision trees and score cards, only differentiate with respects to the (visual) syntax used. Underlying concepts as well as relationship are 100% identical. In addition if-then rules, event, condition& action rules (ECA) and event, condition, action and alternative rules (ECAA) also display high similarities. A closer examination reveals that the only difference between ECA and ECAA is the alternative action concept. Alternative action is a subclass of the "action" concept, which both have the same definition (Knolmayer et al. 2000). Therefore if the "alternative action" would be removed from the conceptual layer but be maintained as a visual element, the decision

language still has the same expressive power. One might argue that symbol synergy is created by this (Moody 2009) but it is quite clear which symbol to use for the first and secondary (alternative action) action.

Table 1. Representational difference analysis business rules languages

Technique	IF-Then		D-Table		D-Tree		ScoreC		ECA		ECAA	
IF-Then	-	-	0	2	0	2	0	2	2	4	3	4
D-Table	0	2	-	-	0	0	0	0	2	4	3	4
D-Tree	0	2	0	0	-	-	0	0	2	4	3	4
ScoreCard	0	2	0	0	0	0	-	-	2	4	3	4
ECA	2	4	2	4	2	4	2	4	-	-	2	4
ECAA	3	4	3	4	3	4	3	4	2	4	-	-

The main difference between the two clusters of languages is caused by two concepts: event and action. An event is “something that ‘happens’ during the course of a process which affects the flow and usually has a cause or an impact and in general requires or allows for a reaction (OMG2011, p113)”. In terms of ECA(A) rules this reaction is an evaluation of predefined conditions (Wu and Dube 2001) leading to a conclusion. Therefore, the event is no actual part of the decision rules but triggers their execution. When executing a business process the event triggering decision business rules is an analytical task (OMG 2011,Zur Muehlen and Indulska 2010).

No general consensus exists regarding the definition of an “action” with respect to ECA(A) rules. This is caused by the adaption of ECA rules in a variety of fields such as personalisation technology, workflow management, rule management and database management (Bailey 2002). In addition one of the first papers (Dayal 1998) introducing the actual ECA mechanism defines an action as something that is executed. However, within BPM literature the action concept is commonly used as the execution of an actual activity (Geppert and Trombos 1998, Van der Aalst et al.2005). Like event, action therefore is no actual part of the decision business rules. Summarising we can conclude that a high synergy between the different modelling languages exists.

So far we did not consider relation theory. In order to do so a second representational difference analysis has to be conducted from a different perspective. The representational difference analysis as well as simplification is summarised in table 2. For now it is sufficient to recognise that the table illustrates the comparison of the different relation views, the interpretation of the formulas will be explained later. Analysis shows that decision tables, decision trees and score cards have the same relational view. This is not surprising considering their meta-models are equal. The remaining three business rules decision languages all have different relational views resulting in four different relationship types. As our goal is to develop a normal form based on relational theory it is important that all languages can be represented by the same relationship (R). Because of practical reasons we do not choose to, and cannot, alter existing business rules languages. A simplification of their relational view is required. Therefore, first all business process element concepts previously indentified are removed from the relationship, see Relational View 2 in table 2. All six relational views are now equal with exception of the conclusion set. Three out of six modelling languages accept only one conclusion set while the others accept multiple. To provide support for every language we remove the possibility to support multiple conclusion sets resulting in Relational View 3. This relationship type therefore is also adopted as our first normal form. Thereby deviating from previous research that allows multiple conclusion facts (Van Thienen and Snoeck1993). In order to further discuss our normalisation procedure and forms first the following concepts need to be further specified: conclusion fact (Cl), condition fact (Cd), business rule and secondary conclusion. Consider the following example, a rule base that is used to decide the specific kind of joint housework for a specific citizen. Two relationships have been defined:

- R1 = (*joint housework*, caring criteria, accommodation criteria, exception criteria, unanswerable presumption)
  - Relational View R1 = (Cl, Cd<sup>1</sup>, Cd<sup>2</sup>, Cd<sup>3</sup>, Cd<sup>4</sup>)
- R2 = (*caring criteria*, financial entanglement, other entanglements, relationship status).
  - Relational View R2 = (Cl, Cd<sup>1</sup>, Cd<sup>2</sup>, Cd<sup>3</sup>)

A relationship is defined on a specific domain of facts which together represent the business rule. A business rule is an actual instantiation of the domain between brackets. A domain of facts contains one fact that is derived

from the other facts within the same domain. Such facts are called conclusion facts (Cl). In the example above, “joint housework” and “caring criteria” are conclusion facts. Facts contributing to the conclusion fact are called condition facts (Cd). Commonly facts from one domain (relation) refer to facts within another domain. To provide a mechanism to address these references the concept of secondary conclusion is introduced. We define secondary conclusion as a fact that represents a conclusion fact in one domain and a condition fact in another, an example of such a fact is “caring criteria”.

Our normalisation procedure is based on database normalisation principles following its general approach (Codd 1970). Therefore as result of the first normalisation form a standard record type is created. This specific record type has already been introduced in previous paragraph:  $R = (Cd^n, Cl)$ . Additional demands are that both Cl and Cd facts must contain a single value. Thus, when the original source is either a decision table, decision tree or score card containing multiple conclusion fact they must be converted to 1<sup>st</sup> normal form. This is realised by duplicating the original business rules the number of times conclusions exist. All of the duplicated rules exist out of all condition and conclusion fields. The difference is that only one of the original conclusion fields is now still a conclusion field while the other are condition fields.

Table 2. Simplification of decision business rule classification

Technique	Original Relation View	Relational View 2	Relational View 3
IF-Then	$R = (Cd^n, Cl)$	$R = (Cd^n, Cl)$	$R = (Cd^n, Cl)$
D-Table	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl)$
D-Tree	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl)$
ScoreCard	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl^n)$	$R = (Cd^n, Cl)$
ECA	$R = (E, C^n, CL, A)$	$R = (Cd^n, Cl)$	$R = (Cd^n, Cl)$
ECAA	$R = (E, C^n, CL, A^n)$	$R = (Cd^n, Cl)$	$R = (Cd^n, Cl)$

After realising a standard representation the relation between conclusion and condition facts has to be normalised. In order to do so partial dependencies and transitive dependencies have to be removed (Codd, 1970, Kent 1984). The latter is realised by applying the 3<sup>rd</sup> normal while 2<sup>nd</sup> normal form deals with the first. In order for a relation to be in second normal form all condition facts must be fully functionally dependent on the conclusion fact and adhere to the 1<sup>st</sup> normal form. Condition facts not fully dependent on the conclusion facts must be deleted or added to another relationship. Second normal form reveals if condition facts are used that actually do not contribute to conclusion. To realise 3<sup>rd</sup> normal form condition facts that are not fully independent on the conclusion fact but another condition fact must be removed and added to a new relation. The new relation contains the removed condition facts as well as the fact that they are the determinants of a conclusion fact. A relationship is established between the two relations by means of a secondary decision. After applying the 3<sup>rd</sup> normal form all specified relationships do not contain any repeating groups, partial dependencies and transitive dependencies. Thereby presenting a language independent view of business rules specifying domain-specific norms for determining a conclusion. The principles described here will be validated in the next section based on existing case study data.

## NORMALISATION OF DECISION BUSINESS RULES: AN EXPERIMENT

Based on existing case study information an experiment has been setup to test and explain the normalisation procedure. The actual case study was executed at a medium sized consultancy organisation. In this experiment we consider the job interview process for employing BPM-consultants; see in figure 2. The first step of the procedure is to determine the scope of the decision to normalise. During the process two analytical tasks are executed, namely “Determine Candidate Profile” and “Discuss Terms of Employment”. In this section we will elaborate on the first. During this activity the candidate will be ranked based on multiple computerised and non-computerised tests s/he undertakes. The test results are input for transactional sequencing business rules that determine whether the candidate is suitable or unsuitable for the job resulting in a termination of the selection procedure or discussing terms of employment.

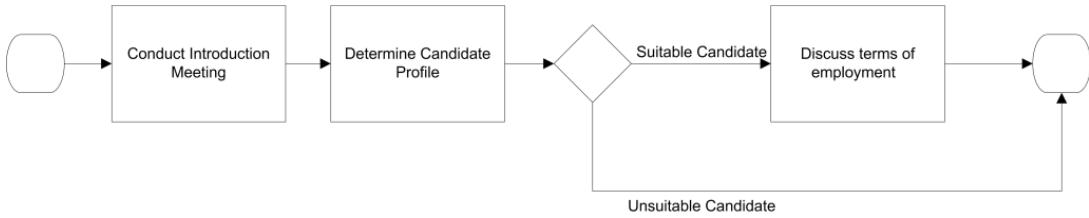


Figure 2: Job interview process for BPM-consultants

After the scope has been determined for the normalisation procedure the next step is the elicitation of the facts and their relationships. This can be done in several ways. First if the organisation already has the condition and facts written down in text or represented in a specific visual syntax they can serve as starting point. If not, backward chaining can be applied to elicitate them. Within our sample case already three decision tables were present, see figure 3.

**Table A: Candidate Ranking**

Conditions				Conclusions
Candidate Cognitive Rating	Candidate Personality Rating	Candidate Compensation Expectation	Candidate Introduction Meeting Rating	Candidate Ranking
Value	Value	Value	Value	Value

**Table B: Candidate Personality Rating**

Conditions						Conclusions	
Candidate Stress Management Rating	Candidate Adaptability Rating	Candidate Intrapersonal Skills	Candidate Interpersonal Skills	Candidate Integrity Rating	Candidate Age	Candidate Maturity Rating	Candidate Personality Rating
Value	Value	Value	Value	Value	Value	Value	Value

**Table C: Candidate Cognitive Rating**

Conditions								Conclusions
Candidate Calculus Test Score	Candidate Reasoning Test Score	Candidate BPMS Knowledge	Candidate MDA Knowledge	Candidate DBA knowledge	Mathematics Cognitive Rating	Candidate Technology Cognitive Rating	Candidate Industry Knowledge	Candidate Cognitive Rating
Value	Value	Value	Value	Value	Value	Value	Value	Value

Figure 3: Decision tables to determine candidate profile

The third step of the procedure is establishing first normal form. First normal form states that every relation can contain only one conclusion fact. In our case study this means that table A and C already are in 1<sup>st</sup> normal form. Table B contains multiple conclusions and therefore needs to be transformed to comply with 1<sup>st</sup> normal form. The transformation exists of creating two identical copies of the tables with two different conclusion facts:

- B.1 = (candidate integrity rating, candidate maturity rating, candidate stress management rating, candidate adaptability rating, candidate intrapersonal skills, candidate interpersonal skills, candidate age, *candidate personality rating*);
- B.2 = (candidate integrity rating, candidate stress management rating, candidate adaptability rating, candidate intrapersonal skills, candidate interpersonal skills, candidate age, candidate personality rating, *candidate maturity rating*).

Second normal form is established when all relationships are in 1<sup>st</sup> normal form and additionally all conditions that are not fully dependent on the conclusion fact are removed. The procedure is executed by determining which of the condition fields are irrelevant when formulating the conclusion and delete them. In our case study this affects relationships B.1 and B.2. Both as a result from applying the 1<sup>st</sup> normal form contain all condition facts from the original relationship B. As it is unlikely that all condition facts contribute to formulating both conclusions the relation has to be investigated. One might argue when transforming the decision table to 1<sup>st</sup> normal form already unnecessary condition facts can be deleted. Although we do not disagree when doing so the 1<sup>st</sup> and 2<sup>nd</sup> normal are

applied simultaneously. Investigation of the relationships reveals that ‘candidate personality rating’ is determined by means of two conclusion facts namely ‘candidate integrity rating’ and ‘maturity rating’. All other condition facts are used to determine ‘candidate maturity rating’ except for ‘candidate age’. Considering that this condition fact also does not affect “personality rating” we remove it all together, resulting in the following relationships:

- B.1.1 = (candidate integrity rating, candidate maturity rating, candidate personality rating);
- B.2.1 = (candidate stress management rating, candidate adaptability rating, candidate intrapersonal skills, candidate interpersonal skills, candidate maturity rating).

Third normal form states a condition fact cannot lead to conclusion about another condition fact. All conditions that are not fully independent on the conclusion fact must be removed and added to a new decision. This procedure is executed as follows. Determine which of the condition fields is not a determinant of the conclusion field but of another condition field. In our case relationship C.1 contains multiple transitive dependencies. After removing the transitive dependencies the following relationships are defined:

- C.1 =(candidate industry knowledge, candidate technology cognitive rating, mathematics cognitive rating, candidate cognitive rating);
- C.2 =(candidate calculus test score, candidate reasoning test score, candidate mathematics cognitive rating);
- C.3 = (candidate BPMS knowledge, candidate MDA knowledge, Candidate DBA knowledge, candidate technology cognitive rating).

All relationships are now in third normal form and specified in a business rules independent language. Also the specified relationships do not contain any repeating groups, partial dependencies or transitive dependencies. The analytical tasks and decision accompanying “Determine Candidate Profile” are shown in figure 4. All conclusion facts are underlined while secondary conclusions are presented in italics.

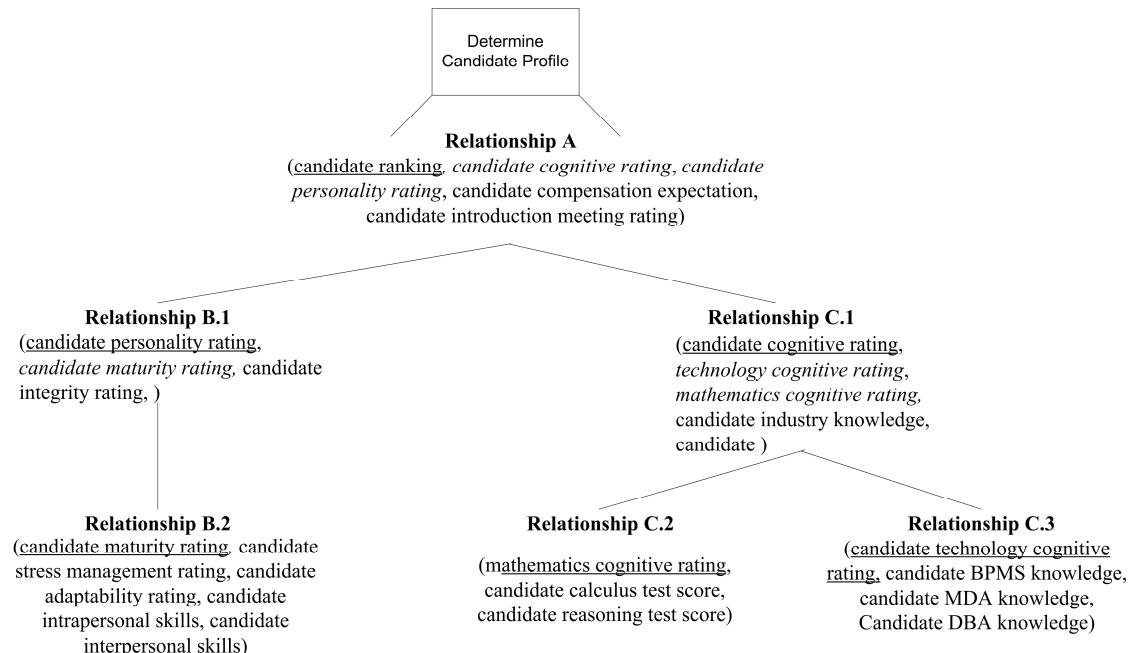


Figure 4: Overview normalised relational view

The second part of the experiment is conducted to test our second hypothesis. To do so all possible update, insertion and delete statements based on the case at hand have been formulated and subjected to all normal forms. Due to space limitation the complete comparison is not provided here, instead a snapshot of the comparison has been added, see table 3. Each row contains the action executed and the number of tuples affected in each of the normal forms.

Table 3. Results of anomalies experiments

Statement	Original	1 <sup>st</sup> NF	2 <sup>nd</sup> NF	3 <sup>rd</sup> NF
-----------	----------	--------------------	--------------------	--------------------

Update existing mathematics cognitive rating value	1	1	1	1
Update existing industry knowledge value	1	2	1	1
Update candidate interpersonal skills	1	1	1	1
Insert new rule for candidate ranking	4	4	4	4
Insert new rule for logic rating	9	9	9	3
Insert new rule for candidate personality rating	8	8	3	3
Delete existing rule candidate cognitive rating	9	9	9	4
Delete existing rule candidate maturity rating	8	8	5	5
Delete existing candidate calculus test score	1	1	1	1

A closer look at the results reveals trends regarding the normalisation procedure. First the update, insertion or deletion of a single tuple affects the same number of tuples in the original as well as all normalised forms. However, one exception can be noted as shown in row three of table 3. This exception is caused when transforming business rules languages that allow multiple conclusions to the 1<sup>st</sup> normal form. During this process the original business rule is duplicated the number of times conclusions exist. All of the duplicated rules exist of (all) condition and conclusion fields. The difference is that only one of the original conclusion fields is now still a conclusion field while the other is a condition field. The removal of the duplicated facts occurs in 2<sup>nd</sup> normal form through which only one tuple is affected again. When inserting and deleting the number of affected tuples decreases, when applying 2<sup>nd</sup> or 3<sup>rd</sup> normal form. The actual tipping point depends on the kind of dependencies that exist between the facts.

## DISCUSSION AND CONCLUSION

Proper control over of anomalies such as insertion, update and deletion is crucial in order to properly structure transactional sequencing business rules that provide analytical activities with guidance. Therefore we defined the following research question: "*How can transactional sequencing business rules guided analytical tasks be normalised such that optimal manageability is realised?*" We developed a normalisation procedure based on representational difference analysis of existing business rules modelling languages, relational theory and database normalisation. Hypothesis 1: "*current decision / transactional sequencing business rules modelling languages can be translated to a unified view by means of relation theory and applied to analytical tasks*" is supported by our proposed procedure. An experiment has been executed showing a decrease in insertion and deletion anomalies when applying our normalised approach, supporting hypothesis 2: "*normalisation of transactional sequencing business rules has a positive effect on the average number of tuples affected by anomalies.*"

We believe that this work represents a further step in research on business rules manageability for analytical tasks. And consequently also a step in the alignment between business rules and business process management has been made. We note however limitations that represent challenges for further research. On the methodological side, we only presented a rather small case study. Although it is expected that with larger rule sets higher savings on anomalies can be accomplished this has not been proven. As we speak, multiple larger case studies are executed to test this hypothesis. Secondly, there are additional questions regarding the economic incentives. When a decrease in anomalies and a more comprehensible rule set is realised a legitimate question is whether the procedure is or will become economically beneficial. For example, from an economic perspective, a rule set only changed twice a year might be better in an un-normalised form.

## REFERENCES

- Arnott, D., and Pervan, G. 2005. "A critical analysis of Decision Support Systems research", *Journal of Information Technology*(20: 2), pp67-87.
- Bailey, J., Poulovassilis A., and Wood, P. 2002. "An Event-Condition-Action Language for XML" In Proceedings of The Eleventh International World Wide Web Conference, pp 24-34.
- Boyer, J., and Mili, H. 2011. *Agile Business Rules Development: Process, Architecture and JRules Examples*. Heidelberg, Germany: Springer.
- Brown, P. 2009. *JBoss Drools Business Rules*. Birmingham, United Kingdom: Packt Publishing.

- Breuker, J., and Van De Velde, W. 1994. *CommonKADS Library for Expertise Modelling: reusable problem solving components*. Amsterdam, The Netherlands: IOS Press.
- Buchanan, B., and Shortliffe, H. 1984. *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Boston, The United States: Addison Wesley.
- Chisholm, M. 2004. *How to Build a Business Rules Engine: Extending Application Functionality Through MetaData Engineering*. San Francisco, The United States: Elsevier.
- Codd, E. 1970. "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM* (13:6), pp. 377-387.
- Cordys. (2010). "Cordys Rules Management Enabling Consistent Organization Policies Managed by the Business." Retrieved 10 December, 2010, [http://www.cordys.com/cordyscms\\_com/rules\\_management.php](http://www.cordys.com/cordyscms_com/rules_management.php)
- Dayal, U., Buchmann, P. and McCarthy, R. 1988. "Rules are objects too: A knowledge model for an active, object-oriented database system," *Lecture Notes in Computer Science* 334, pp 129-143.
- Dominguez, A. 2009. "Assignment Preconditions." Retrieved 10 December, 2010, [http://wiki.bizagi.com/en/index.php?title=Assignment\\_Advanced\\_Features#Assignment\\_Rule](http://wiki.bizagi.com/en/index.php?title=Assignment_Advanced_Features#Assignment_Rule)
- Gottesdiener, E. 1997. "Business Rules Show Power, Promise," *Application Development Trends*(4:3), pp. 36-42.
- Greppert, A. and Dimitrios, T. 1998. "Technical Report 96.05: Event-based Distributed Workflow Execution with EVE" Retrevied 2 December, 2010,<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.5479>
- Graham, I. 2006. *Business Rules Management and Service Oriented Architecture*. New York, TheUnited States: Wiley.
- Green, P. and M. Rosemann. 2004. "Applying Ontologies to Business and Systems Modeling Techniques and Perspectives: Lessons Learned," *Journal of Database Management* (15:2), pp. 105-117.
- Heimrich, T., and Günther, S. 2003. "Enhancing ECA Rules for Distributed Active Database Systems," *Lecture Notes in Computer Science* 2593, pp. 199-205.
- Hevner, A., March, S., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *Management Information Systems Quarterly* (28:1), pp. 75-105.
- Hevner, A. and Chatterjee, S. 2010. *Design Science in Information Systems Research: Theory and Practice*.Heidelberg, Germany: Springer.
- Hubank, M. and Schatz, D. 1994. "Identifying differences in mRNA expression by representational difference analysis of cDNA," *Nucleic Acids Research*(22:5), pp. 5640 - 5648.
- Kent, W. 1983. "A simple guide to five normal forms in relational database theory," *Communications of the ACM*(26:2), pp. 120-125.
- Knolmayer, G., Endl, R., and Pfahrer. M. 2000. "Modeling Processes and Workflows by Business Rules," *Lecture Notes in Computer Science* 1806, pp. 16-29
- Kohavi, R. 1995. "The power of decision tables," *Lecture Notes in Computer Science* 912, pp. 174-189.
- Ligêza, A. 2006. *Logical Foundations for Rule-Based Systems*.Heidelberg, Germany: Springer.
- Moody, D. 2009. "The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering," *Software Engineering: IEEE Transactions*(35:6), pp. 756-779.
- Morgan, T. 2002. *Business Rules and Information Systems*.Indianapolis, The United States: Pearson Education.
- Morrow, D., Antman, E., Charlesworth, A., Cairns, R., Murphy, S., Lemos, J., Giugliano, R., McCabe, C., and Braunwald, E. 2000. "TIMI Risk Score for ST-Elevation Myocardial Infarction: A Convenient, Bedside, Clinical Score for Risk Assessment at Presentation," *Circulation* (10:2) pp. 2031-2037.
- OMG. (2011). Business Process Model and Notation. Retrieved 4 March, 2011, <http://www.omg.org/spec/BPMN/2.0/PDF>.
- Pegasystems. (2011). Pega Decision Management Optimize Your Business Processes with Real-Time Decisioning. Retrieved 2 June, 2011,<http://www.pega.com/products/decision-management/business-rules>
- Quinlan, J. 1986. "Induction of decision trees," *Machine Learning*(1:1), pp. 81-106.

- Qumer, A. and Henderson-Seller, B. 2008.“A framework to support the evaluation, adoption and improvement of agile methods in practice,” *Journal of Systems and Software* (81:11), pp. 1899-1919.
- Rivest, R. 1987. Learning decision lists. *Machine Learning*(2).pp. 229-246.
- Ross, R. 2003.*Principles of the Business Rule Approach*. Boston, The United Sates: Addison Wesley.
- Ross, R. 2009.*Business Rule Concepts*. Houston, The United States: Business Rule Solutions, LLC.
- Schacher, M. andGrässle, P. 2006. *Agile Unternehmendurch Business Rules:Der Business Rules Ansatz*. Heidelberg, Germany: Springer.
- Van Thienen, J. And Snoeck, M. 1993.“Knowledge factoring using normalisation theory”, *International Symposium on the Management of Industrial and Corporate Knowledge*, pp. 42-52.
- Van der Aalst, W., Ter Hofstede, A., and Weske, M. 2003. “Business Process Management: A Survey”, *Lecture Notes in Computer Science* 2678, pp. 1-12.
- Van der Aalst, W., Weske, M. and Grünbauer, D. 2005. “Case Handling:A New Paradigm for Business Process Support,”*Data and Knowledge Engineering*(53:2), pp. 129–162.
- Von Halle, B. 2001.*Business Rules Applied: Building Better Systems Using the Business Rules Approach*. New York, The United States: Wiley.
- Zoet, M., Ravesteyn, P., and Versendaal, J. 2011. “Alignment of Business Process Management and Business Rules” *In Proceedings of European Conference on Information Systems (ECIS) 2011*, paper 34, pp. 1-12.
- Zur Muehlen, M. and Indulska, M. 2010.“Modelling languages for business processes and business rules: A representational analysis,”*Information Systems*(35:4),pp. 379-390.

## COPYRIGHT

Martijn Zoet, Pascal Ravesteyn and Johan Versendaal © 2011. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the explicit permission of the authors.