Making Business Rules Operational - Business Rule Generation and Quality

Knut Hinkelmann
Realizing Business Rules

- There are various approaches to operationalize rules, e.g.
  - Rule engines: specialized programs designed to execute rules
  - Program code: encapsulating a rule or rule set in a function
  - Databases: business rules could implement integrity constraints, stored procedures, or triggers
  - Workflow Management Systems: rules are mostly associated with branching points
Business Rules Technology

There are different types of Business Rules technology

- **Business Rules Management System (BRMS):** a software system used to define, deploy, execute, monitor and maintain business rules. It includes
  - A *repository*, allowing business rules to be stored
  - A business rules engine, allowing applications to invoke business rules and execute them in a runtime environment
  - Maintenance tools, allowing both technical developers and business experts to define and manage business rules, e.g. supporting simulation, testing, quality checking

- **Business Rules Discovery:** Automatically finding rules (e.g. in form of decision trees, decision tables) by using data mining techniques
From Textual Rules to Formal Structures: Low Technology Rule Definition

Current generation of tools for rule definitions
- Business Analyst defines rules on informal level (→ easy to read)
- Translation to formal structures by a Designer
- A Developer implements the business rules making them operational in systems

Problem: Transformation can cause errors

(Morgan 2002, p. 64)
From Textual Rules to Formal Structures: The long-term Objective

- The ultimate goal would be to generate code fully automatically.

(Morgan 2002, p. 64f)
Business Rules Management Systems

Editing JRules in the ILOG Rule Builder

If
  the shopping cart value equals : $ <enter a value> [±]

Then
  <select an action>
  equals
  does not equal
  is greater than
  is greater than or equal to
  is less than
  is less than or equal to

NewRule *
Decision Table

- Decision table are a popular way to represent multiple decision rules:

- General structure of a decision table

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>Condition N</th>
<th>Action 1</th>
<th>Action 2</th>
<th>Action N</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Y</td>
<td>Y</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Example

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 5 years</td>
<td>✓</td>
</tr>
<tr>
<td>&gt;= 5 and &lt; 18</td>
<td>x</td>
</tr>
<tr>
<td>&gt;= 18 and &lt; 55</td>
<td>x</td>
</tr>
<tr>
<td>no concession card</td>
<td>x</td>
</tr>
<tr>
<td>&gt;= 55</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Admission</td>
</tr>
<tr>
<td>$8.00</td>
</tr>
<tr>
<td>$12.00</td>
</tr>
<tr>
<td>$6.00</td>
</tr>
</tbody>
</table>
The Visual Rules Editor

For screencasts on Visual Rules see http://www.visual-rules.com/screencasts-demos-business-rules.html#
Finding Rules

Approaches for finding rules:

- **Static analysis**
  - best approach when relevant documentation is available
  - careful checking of source documents for potential rules

- **Interactive sessions**
  - bring together analysts and business specialists in *structured interviews* or *analysis workshops*
  - applicable where business knowledge is not readily available in a documented form

- **Automated rule discovery**
  - find rules through machine analysis (*data mining, code analysis*)
  - provided that suitable source data can be made available

for more details see (Morgan 2002, pp. 110-121)
Clarity of Business Rules

- Business Rule statements must be in a form that the business owner can immediately accept them as valid or reject as invalid.

- Thus, Business Rules are a series of simple statements about the business with the following characteristics:

  Atomic: can't be broken down any further without losing information

  Unambiguous: have only one, obvious, interpretation

  Compact: typically, a single sort of sentence

  Consistent: together, they provide a unified and coherent description

  Compatible: use the same terms as the rest of the business model

(Morgan 2002, p. 61)
Tips on Rule Construction

- Some common problems in rule construction can be avoided following some general recommendations.

- Examples:
  - Use a fact model so that rules can be related to other parts of the business model
  - Split complex rules into several simple rules if possible
  - Whenever possible avoid using plurals as terms of rules
  - Avoid ambiguous states
  - …

- More tips with detailed descriptions can be found in (Morgan 2002, pp. 79-90).
Controlling Rule Quality

- Quality control mechanisms that can be applied during rule development
- **Walkthroughs**: Workshop-style review sessions
  - as soon as enough rules are defined to support a business scenario
- **Inspections**: more formal type of review
  - involving representatives from many business areas
  - used mostly at major milestones
- **Testing**: ensure a clear understanding of complex rule sets
  - understand the logic of whole sets of rules
  - applying a series of specific test cases to a trial implementation of the rule set
Typical Assessment Activity Pattern

(Morgan 2002, p. 133)
Reviewing Rules: What to look for

- Look for problems of rules, e.g. rules that are
  - malformed: rules that don't conform to standards or preferred rule patterns
  - incomplete: a situation is not properly covered by the rules
  - inconsistent: leading to ambiguous results with different rules
  - redundant: serve no business purpose of are covered by another rule
  - use terms not properly rooted in the supporting fact model
General Structure of a Review

(Morgan 2002, p. 135)
## Quality Controls

<table>
<thead>
<tr>
<th>Feature</th>
<th>Reviews</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Walkthroughs</td>
<td>Inspections</td>
</tr>
<tr>
<td><strong>What's examined?</strong></td>
<td>Rule population, possibly incomplete</td>
<td>Complete rule population</td>
</tr>
<tr>
<td><strong>When used?</strong></td>
<td>As often as practical, starting as soon as a reasonable body of rules is assembled</td>
<td>Toward the end of a project phase, before a rule population is released</td>
</tr>
<tr>
<td><strong>What's checked?</strong></td>
<td>Rule clarity and business relevance</td>
<td>Rule clarity and business relevance, along with consistency of rule population</td>
</tr>
<tr>
<td><strong>Focus defined by</strong></td>
<td>Selected business scenarios</td>
<td>Business scope of rule population</td>
</tr>
<tr>
<td><strong>Purpose of meeting</strong></td>
<td>Work through rules and raise actions</td>
<td>Work through pre-prepared comments and consolidate into actions</td>
</tr>
<tr>
<td><strong>Results on file</strong></td>
<td>Observations and actions from each walkthrough, checked as completed</td>
<td>Observations and actions from each inspection, checked as completed</td>
</tr>
</tbody>
</table>

(Morgan 2002, p. 167)
References

For more details on business rules quality see