2nd International Workshop on Decision Mining & Modeling for Business Processes (DeMiMoP’14)

In conjunction with BPM 2014, Eindhoven,
Sept 8, 2014
Chairs

Prof. dr. Jan Vanthienen (corresponding organizer)
Department of Management Informatics
KU Leuven
Naamsestraat 69 - bus 3555, 3000 Leuven, Belgium
Tel: +32 16 326878
E-mail: jan.vanthienen@kuleuven.be
URL: http://www.econ.kuleuven.be/jan.vanthienen

Prof. dr. Qiang Wei
Department of Management Science and Engineering
School of Economics and Management
Tsinghua University
443 Weilun Building, Tsinghua University, Haidian, 100084 Beijing, China
Tel: +86 (10) 62789824
E-mail: weiq@sem.tsinghua.edu.cn
URL: http://www.sem.tsinghua.edu.cn/en/weiq

Prof. dr. Bart Baesens
Department of Management Informatics
KU Leuven
Naamsestraat 69 - bus 3555, 3000 Leuven, Belgium
Tel: +32 16 326884
E-mail: bart.baesens@kuleuven.be
URL: http://www.bartbaesens.com

Prof. dr. Guoqing Chen
Department of Management Science and Engineering
School of Economics and Management
Tsinghua University
258 Weilun Building, Tsinghua University, Haidian, 100084 Beijing, China
Tel: +86 (10) 62772940
E-mail: chengq@sem.tsinghua.edu.cn
URL: http://www.sem.tsinghua.edu.cn/en/chengq
Goals of the workshop

• to examine the relationship between decisions and processes;
• to enhance decision mining based on process data;
• to examine decision goals, structures, and their connection with business processes, in order to find a good integration between decision modeling and process modeling;
• to study how different process models can be designed to fit a decision process, according to various optimization criteria, such as throughput time, use of resources, etc.;
• to show best practices in separating process and decision concerns.
Overview

• Decisions and processes
• Decision model & notation
• Decision processes
• Intelligent BPM
Introduction

Decisions and processes
Level 0: Rules/decisions are not processes

> rules, decisions “hard-coded” in the process model
This is more flexible

- Business logic rules separated from the process, reduce the process to its essence
- Because rules change more often than processes

- Thinner processes
What should be in the process model?

Exceptions?
Timers?
Happy path?
Decisions?
Roles?
Messages?
Notifications?
Triggers?
Conditions?

…
Level 1: Processes contain decisions

- **When to accept/refuse a claim?**
  Depends on:
  - Customer history
  - Format requirements
  - Timing constraints
  - Type of contract
  - ...

Decision Management
Analytics
Decision Modeling
Decision Mining
Business Modeling

Model Driven Architecture (MDA)

- Business
  - BMM
  - SBVR
  - Business Vocabulary
- Platform Independent Model
  - OSM
- Platform Specific Model
  - BPDM + BPMN
- Computation Independent Model
  - DMN
  - UML
  - BPM
  - BPMN
  - W3C
  - OWL
  - RuleML
  - PRR
- "S-beaver"

Decision Mining & Modeling - DeMiMoP 14
Decision model & notation
DMN Motivation

1. Decisions are important for Business
2. Modeling decisions using BPMN is not ideal (i.e. hardcoding)
3. Decisions/rules are not only in (automated) processes, also in vocabulary, and manual processes
4. Relationships between Decisions, Rules are often obscured
5. Distinction between Business Rule Management Systems (BRMS), Notation Methods, Modeling tools, Rule engines (BREs)
6. Tabular methods are increasingly being used in BRMS, BREs
Decision and Process Model

Business Process Model (BPMN)

Decision Model (DMN)

Routing table

Routing

Application risk

Eligibility

Application risk score model

Eligibility rules

Decision Requirements Level

Decision Logic Level

<table>
<thead>
<tr>
<th>p</th>
<th>Employment status</th>
<th>Country</th>
<th>Age</th>
<th>Eligibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNEMPLOYED</td>
<td></td>
<td></td>
<td>INELIGIBLE</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>not(UK)</td>
<td></td>
<td>INELIGIBLE</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td></td>
<td>&lt;18</td>
<td>INELIGIBLE</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td></td>
<td></td>
<td>ELIGIBLE</td>
</tr>
</tbody>
</table>
New BPMN 2.0 Activity Types

applicant type rules

determine applicant type
DMN levels

• Decision Requirements Level
  o Decision Requirements Diagram

• Decision Logic Level
  o Decision Tables
    • Single Hit: return one outcome
    • Multiple Hit: return list of all applicable outcomes
  o Other Functions
Decision Requirements Diagram
Model of the decision
The process
Example: Modeling the decision

Naturalization is the acquisition of citizenship by somebody who was not a citizen of that country when born. Not everyone can request naturalization and the decision whether someone can apply for a new nationality is restricted by several requirements. The requirements for the Belgian citizenship are formulated by the following decisions:

- Is the applicant of legal age?
- Does the applicant have a legal residence? (What does it mean to have a legal residence?); Has Belgium been the applicant’s main country of residence?
- Can the applicant show that he is socially and culturally integrated?
Decision goal network

- Decision on naturalization application (A)
  - Decision on appropriateness of residence status (B)
    - Collection of ID-card data (D)
    - Evaluation of residency conditions (E)
  - Assessment of the cultural and social integration (C)
Decision table structures

From DB, Ask user, ...

Customer.Type?

1. Account.Age (years)
2. Account.Turnover (T)
3. Customer.Type

Orders
1. Customer.CreditLimit
2. Customer.Type
3. Stock.Quantity
1. Execute Order
2. Refuse Order
3. Put on Waiting List

Customer.CreditLimit
<table>
<thead>
<tr>
<th>Ok</th>
<th>Not Ok</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Customer.Type
<table>
<thead>
<tr>
<th>Good</th>
<th>Not Good</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stock.Quantity
<table>
<thead>
<tr>
<th>Ok</th>
<th>Not Ok</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Execute Order
| - | - | x | - | - |

Refuse Order
| - | - | - | x |

Wait List Order
| - | x | - | x |

Order.Quantity
<table>
<thead>
<tr>
<th>Order Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;10</td>
</tr>
<tr>
<td>&gt;=10 and &lt;15</td>
</tr>
<tr>
<td>&gt;=50 and &lt;100</td>
</tr>
<tr>
<td>&gt;=100</td>
</tr>
</tbody>
</table>

Order Distance
| - |

Discount
<table>
<thead>
<tr>
<th>Transport</th>
<th>Bill Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Road</td>
</tr>
<tr>
<td>10%</td>
<td>Road</td>
</tr>
<tr>
<td>5%</td>
<td>Road</td>
</tr>
<tr>
<td>2%</td>
<td>Road</td>
</tr>
<tr>
<td>10%</td>
<td>Railway</td>
</tr>
</tbody>
</table>
DMN identifies different table types, indicated by the first letter:

- **(default) unique hit** tables: every input case is included in one rule only. There is no overlap between rules.
- **any hit** tables: every input case may be included in more than one rule, but the outcomes are equal. Rules are allowed to overlap.
- **priority hit** tables: multiple rules can match, with different outcome values. This policy returns the matching rule with the highest output value priority (e.g. highest discount).
- **first hit** tables: multiple (overlapping) rules can match, with different outcome values. The first hit by rule order is returned (and evaluation can halt). The table is hard to validate manually and therefore has to be used with care.

DMN does not prescribe a specific form (although there is a default), but at least makes sure there is no ambiguity.
## Applicant Risk Rating

<table>
<thead>
<tr>
<th>Applicant Risk Rating</th>
<th>Applicant Age</th>
<th>Medical History</th>
<th>Applicant Risk Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Applicant Age</td>
<td>Medical History</td>
<td>Applicant Risk Rating</td>
</tr>
<tr>
<td>1</td>
<td>&gt; 60</td>
<td>good</td>
<td>Medium</td>
</tr>
<tr>
<td>2</td>
<td>bad</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>[25..60]</td>
<td>bad</td>
<td>Medium</td>
</tr>
<tr>
<td>4</td>
<td>&lt; 25</td>
<td>good</td>
<td>Low</td>
</tr>
<tr>
<td>5</td>
<td>bad</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>

## Applicant Risk Rating

<table>
<thead>
<tr>
<th>Applicant Age</th>
<th>&lt; 25</th>
<th>[25..60]</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical History</td>
<td>good</td>
<td>bad</td>
<td>-</td>
</tr>
<tr>
<td>Applicant Risk Rating</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

## Applicant Risk Rating

<table>
<thead>
<tr>
<th>Applicant Age</th>
<th>&lt; 25</th>
<th>[25..60]</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical History</td>
<td>good</td>
<td>bad</td>
<td>-</td>
</tr>
<tr>
<td>Low</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Medium</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>High</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
So what issues does DMN solve?

- Separating decisions and their logic from the process simplifies the process
- There was no standard model for decisions
- Standardizing decision table semantics ensures that tabular methodologies are unambiguous
- Separating the decision from the decision logic allows to model decision relations (even if not all logic is expressed in tables)
- Defining a strict notation for decision tables (and a simple expression language), allows business people to truly model and maintain decision logic
Decision Table Methodology

- Guidelines for composing effective decision tables (form, structure, meaning, …).
- Developing a hierarchical structure of tables that serves as a functional solution to the problem.
- Definition of a working subset of decision tables (completeness, no inconsistencies, no redundancies, no overlapping rules, no ELSE-rule, …).
- A simple 8 step method to construct good decision tables.
- A transition from the specification to design, implementation and maintenance.

So, problem solved?

• We keep modeling the process as we did
• and put some decisions in lower level decision models

• So now we can model the global process and make abstraction of some details

• Wait a minute!
Level 2: sometimes the process is a decision

Hard-coding the decision process

- the decision process
- the data acquisition process
- the decision sequence

(Bruce Silver)
Decision processes
Decision goal network

The main decision Z has only one incoming edge, this signals that the decision is dependent on the value of all lower level decisions A, B, C and D. Decision B on the other hand has three incoming edges, this means that it only needs the outcome of one lower level decision.
From decisions to processes

- Start each individual decision activity as soon as all its preconditions are fulfilled.
- Avoid superfluous decision activities (unnecessary work).
- Group customer contacts.
An example process model
Evaluating this model

- **Customer perspective**: the applicant must come to the administration twice, for ‘assess the integration of the applicant’ and ‘check ID card for foreigners’.

- **Business process behavioral perspective**: starting with a labor-intensive activity as ‘assess the integration of the applicant’ is not optimal since the application could be easily rejected when Belgium is not the country of residence (simple check in the Register).

- **Organizational perspective**: the number of handovers can be halved by letting the ‘check ID card for foreigners’ succeed the ‘assess the integration of the applicant’ immediately.

- **Informational perspective**: the current activity sequencing imposes a double consultation of the Register of Births in some cases, while all necessary information could be easily at one point in time.

- **External environmental perspective**: assessing legal age might require intervention from an embassy. The number of contacts with embassies should be limited.
Other models
Criteria to choose between different models

Criteria distinguished by (Reijers, 2005):

- **Customer**: This criterium improves the relations and contacts with the customer such that the contacts happen in an efficient way and customer friendly manner. An example: reducing the customer touchpoints, by having all information available upfront.

- **Business process operation**: This criterium implements the workflow in an efficient way. An example: eliminating redundant activities.

- **Business process behavior**: This criterium optimizes the time aspect of the execution. An example: moving activities to the front that are likely to fail.

- **Organization**: This criterium optimizes the organizational structure and the involved resources. An example: assigning the same worker to a case for the whole process. Because the worker has prior knowledge about the case he doesn’t need to get accustomed to the case every time.

- **Information**: This criterium uses best practices for the usage of information in business processes. An example: minimize information requests by checking incoming and outgoing information.

- **External environment**: This criterium improves the collaboration and communication with third parties. An example: using standardized interfaces with customers and criteria.

Level 3: Business has many rules and decisions

- Rules about
  - Claim handling decisions
  - Who makes decisions
  - Reusable rules across processes
  - General processes and run-time exceptions
  - Process steps
  - Event handling decisions
  - Authorizations
"Today processes are mostly modeled and structured for expected conditions, but there are dangers in pursuing that strategy exclusively.

I would propose we need to start thinking about unexpected exceptions and processes that are less structured (unstructured) to adapt to change and to include work that is more fluid. There are a whole lot of benefits in dealing with unstructured processes.

Process may have to flex as goals change, so coupling dynamic goals with dynamic and unstructured processes will allow the change of process composition, sequence and outcomes."

Process modeling & enforcement

static

procedural

Very strict - structured with some exceptions

Hard-coded rules and processes:
For things that never change

Hard-coded processes:
For high-volume processes that rarely change.
No exceptions. Decision services in rules

More flexibility in the process:
For high-volume processes that contain many alternative scenarios

Derived execution scenario:
For stable processes with lots of rules

dynamic

declarative

- adaptive with some strict subsets

- adaptive case management

(ACM)

- adaptive

Run-time execution scenario:
For highly volatile processes with lots of rules
Thank you