Decision-aware Business Processes

Knut Hinkelmann
Combining Structured Processes and Knowledge Work

We deal with two special kinds of modeling knowledge work:

- **Business Rules**
  - Modeling knowledge work as a set of rules, e.g. decision rules
  - Corresponds to the Transaction and Expert Model of Davenport (2010)

- **Case Management**
  - Collaborative process
  - No predefined process flow
  - Can occur as a (ad-hoc) subprocess or be a process of its own.
Combining Structured Processes and Knowledge-Intensive Tasks

- A process model represents the flow of a process (= process logic)
- Business Logic should be separated from the process logic.
- In BPMN there is the task type "Business Rules". This type can be used to label knowledge-intensive tasks.
- Decision tasks are a special kind of knowledge-intensive tasks
**Decision Tasks in Business Processes**

- A **decision task** is a knowledge-intensive task in which some decision is made.
- Decision tasks are based on decision logic.
- Decision tasks can provide data for gateways.
  - At the gateway only the result of the decision should be tested (for the selection of the path) not the criteria for the decision.

![Diagram: Decision Flowchart]

- **Check Eligibility**
- **Applicant eligible?**
  - Yes: **pay reimbursement**
  - No: **reject application**

**Decision:** Is the applicant eligible?

**Decision:** Which reimbursement scheme is appropriate in this specific case?
Basic Elements of Operational Business Decisions

■ A decision is characterized by a question, for example:
  ♦ Should the insurance claim be accepted, rejected or examined for fraud?
  ♦ Which resource should be assigned to this task?
  ♦ Which service should be used to ship this package?

■ A potential outcome is some result, conclusion, or answer that might be deemed appropriate for a case. Examples:
  ♦ some form of yes/no (e.g. eligible/non-eligible)
  ♦ some quantities (e.g. dollar amounts)
  ♦ some categories (e.g. silver, gold, or platinum customer)
  ♦ some real-world instances (e.g. software product to be purchased)
  ♦ some course of action (e.g. on-site visit, teleconference, email)

■ A case is some particular matter or situation arising in a day-to-day business activity and requiring consideration

■ The outcome is the result, conclusion, or answer for a given case

■ The business logic that is used for decision making is called decision logic (the set of all decision rules selecting a decision outcome) (Ross 2011, p. 152f)
Examples for Elements of Business Decisions

- Process: Handling auto insurance applications
- Decision Task: Check Eligibility of Applicant
- Potential outcomes: "yes" and "no" (i.e. eligible/non-eligible)

Case: John Smith applies for an auto insurance

Decision Task: Check Eligibility

Outcome: John Smith is eligible for auto insurance

How to represent Decision Logic?

(Ross 2011, p. 152f)
Representation of Decision Rules

■ There are a variety of ways to represent decision rules, e.g.
   ♦ Semi-formal description
     ● The reimbursement is 90% if the patient visited a doctor's office and the physician was present
   ♦ Decision Table

<table>
<thead>
<tr>
<th>Conditions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Type of visit</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>2. Participating Physician?</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Reimburse 50%</td>
</tr>
<tr>
<td>2. Reimburse 70%</td>
</tr>
<tr>
<td>3. Reimburse 90%</td>
</tr>
<tr>
<td>4. Impossible or no reimbursement</td>
</tr>
</tbody>
</table>

Reimbursement depends on whether the patient visited the doctor's office (D), a hospital (H) or a lab (L) and whether the Doctor is a Participating Physician.

Each column represents a rule.

http://web.sxu.edu/rogers/sys/decision_tables.html
Declarative Representation of Decision Logic

- Decision logic should be represented declaratively
- A declarative representation of Decision Logic
  - specifies the conditions on which a decision is made
  - does not specify how the conditions are tested, in particular it does not specify the order in which conditions are tested
- Examples of declarative representations of decision logic
  - decision tables
  - business rules
  - The Decision Model TDM (von Halle & Goldberg 2010)
  - Decision Model and Notation (OMG 2014)
  - Q-Charts (Ross 2011)
**Decision-Aware Process Models: Managing Process Logic and Decision Logic Separately**

- The process model contains the process logic.
- The decision logic can occur in different forms:
  - implicit in head of people
  - as text (e.g. guidelines)
  - as rules
  - as formal decision model
- Decision logic can be assigned to tasks in the process model (in particular to knowledge-intensive tasks)

**Process Logic**

**Business Logic / Decision Logic**
The Decision Model TDM
The Decision Model

The Decision Model was developed by Barbara von Halle and Larry Goldberg (2009)

Objective:
- a rigorous, repeatable, and technology-independent model of business logic that is simple to create, interpret, modify, and automate

The Decision Model is a template for perceiving, organizing, and managing the business logic behind a business decision.

It is a declarative representation of decision logic
- *specifies* the conditions on which a decision is made
- *does not specify* how the conditions are tested, in particular it does not specify the order in which conditions are tested
Decision Model Elements

A Decision Model has two different kinds of diagrams:

Decision Model Diagram

Rule Family Table

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Student Loans</td>
<td>Person Business Loans</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Decision Model Diagrams

- The root of a Decision Model diagram (its start) is an octagonal shape that represents the entire business decision.
  - It is this shape that relates to tasks within business process models.

- The other nodes in the Decision Model diagram represent Rule Families.

(von Halle & Goldberg 2010, p. 26f)
Decision Model Diagram

The Decision Model Diagram represents Rule Family Tables.

(von Halle & Goldberg 2010, p. 29)
Rule Family: Basic Element of the Decision Model

- Rule Family is a two-dimensional table relating conditions to one—and only one—corresponding conclusion.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Employment History</td>
<td>Person Likelihood of Defaulting on a Loan</td>
</tr>
<tr>
<td>Person Mortgage Situation</td>
<td></td>
</tr>
<tr>
<td>Person Miscellaneous Loans Assessment</td>
<td></td>
</tr>
</tbody>
</table>

Column headings: names of facts being tested

- Value of column heading

- Operator

- Condition

(von Halle & Goldberg 2010, p. 18f)
Rule Family Tables are Decision Tables

- A Rule Family table is a kind of decision table
  - In a Rule Family Table each row represents a rule
  - In a decision table each column represents a rule

- A Decision Model is a structured collection of decision tables

- There are some specialties:
  - A Rule Family must only have one conclusion column
  - Inferential relationships between Rule Family are made explicit in a Decision Model diagram

(von Halle & Goldberg 2010, p. 25)
Rule Family: Basic Element of the Decision Model

- A Rule Family node has three parts:
  - The name is the conclusion of the Rule Family
  - Inferred conditions: There are Rule Families with these names
  - Basic conditions: There are no Rules Families with these names

(von Halle & Goldberg 2010, p. 18f)
This Condition column is part of a logical expression interpreted as “If/when the Policy Renewal Override is Yes”

The conclusion column is part of a logical expression interpreted as “then the Policy Renewal Method is Automatic”

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Manual Policy Override</th>
<th>Policy Tier Within Bounds</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>is</td>
<td>Yes</td>
<td>is Manual Renewal Process</td>
</tr>
<tr>
<td>2</td>
<td>is</td>
<td>No</td>
<td>is Manual Renewal Process</td>
</tr>
<tr>
<td>3</td>
<td>is</td>
<td>Yes</td>
<td>is Automatic Renewal Process</td>
</tr>
</tbody>
</table>

A discrete business logic instance is a single row in the Rule Family table

This condition column is also part of a logical expression interpreted as “If/when the Policy Tier Within Bounds is Yes”

(von Halle & Goldberg 2010, p. 29)
Translating a Rule Family into Natural Language

It is possible to convert each row in a Rule Family into a sentence that sounds natural to a business audience.

Possible Conversions

- If/when Person Employment History is Poor and Person Mortgage Situation is Poor and Person Miscellaneous Loans Assessment is High, then the Person Likelihood of Defaulting on a Loan is High.
- A Person with Poor Employment History and Poor Mortgage Situation and High Miscellaneous Loans Assessment has a High Likelihood of Defaulting on a Loan.
- It is obligatory that the Person Likelihood of Defaulting on a Loan is High if the Person Employment History is Poor and the Person Mortgage Situation is Poor and the Person Miscellaneous Loans Assessment is High.
A Rule Family represents all Rules for one Conclusion

The Decision Model has only one Rule Family for each type of conclusion column, i.e. all rules for a conclusion are in one table.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Manual Policy Override</th>
<th>Policy Tier Within Bounds</th>
<th>Policy Renewal Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>is</td>
<td>Yes</td>
<td>is Manual Renewal Process</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>is</td>
<td>is Manual Renewal Process</td>
</tr>
<tr>
<td>3</td>
<td>is</td>
<td>No</td>
<td>is Automatic Renewal Process</td>
</tr>
</tbody>
</table>

This condition column is also part of a logical expression interpreted as “If/when the Policy Tier Within Bounds is Yes”

(von Halle & Goldberg 2010, p. 29)
Rule Pattern

- A set of Rule Family rows with a common set of populated condition cells is called a Rule Pattern.

- The following Rule Family represents two rule patterns

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Person Student Loans</th>
<th>Person Business Loans</th>
<th>Person Customer Status</th>
<th>Person Miscellaneous Loans Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Is</td>
<td>Yes</td>
<td>Is not</td>
<td>Current customer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low Risk</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>Medium Risk</td>
</tr>
</tbody>
</table>

1. The 1st and 3rd rule have conditions for the fact types "Person Student Loans" and "Person Customer Status"

2. The 2nd and 4th rule have conditions for the fact types "Person Business Loans" and "Person Customer Status"

(von Halle & Goldberg 2010, p. 24)
Two dependend Rule Families

- Conditions of one rule family can depend on another rule family
- Example: Person Employment History in the first rule family depends on
  - Person Years at Current employer &
  - Person Number of Jobs in Past Five Years

(von Halle & Goldberg 2010, p. 23)
The Decision Model diagram shows the inferential relationships between Rule Families.

Solid lines between Rule Family nodes represent inferential nodes:
- The name of the node at the end with the dot occurs as condition in the other node.

- Leave nodes in a Decision Model diagram to not have inferred conditions.

(von Halle & Goldberg 2010, p. 26f)
The Rule family directly connected to the business decision shape is called the “Decision Rule Family”, its conclusion is the conclusion sought by the entire Decision Model.

The Decision Model diagram begins with an octagonal shape that represents the entire business decision. The other shapes in the Decision Model diagram represent Rule Families. This diagram has 6 Rule Families.

A Decision Model diagram begins with an octagonal shape that represents the entire business decision. The other shapes in the Decision Model diagram represent Rule Families. This diagram has 6 Rule Families.

The name of each Rule Family is its conclusion column heading.

Inferring value for the first Condition of the Decision Rule Family

Inferring value for the second Condition of Decision Rule Family

Conditions based on facts

The Decision Model diagram begins with an octagonal shape that represents the entire business decision.
The solid line terminated by the dot connects Rule Families that have an inferential relationship: The conclusion of one Rule Family is used as a condition in another.

The dotted line: The conditional columns are shown below the solid line and above the dotted line.

Inferred Conditions

Conditions based on facts

(von Halle & Goldberg 2010, p. 28)
The Rule Family directly connected to the business decision shape is called the “Decision Rule Family”.

The Name of a Rule Family is the conclusion column heading.

The solid line terminated by the dot connects Rule Families that have an inferential relationship. In this case the condition column “Policy Renewal Override” in the Decision Rule Family has an inferential relationship with the conclusion column of the “Policy Renewal Override” Rule Family.

There are condition columns that do not serve as conclusion columns in another Rule Family. These condition columns will be populated by known fact values (e.g., persistent data).

The labels below the solid line but above the dotted line denote condition column headings that serve as conclusion column headings in another Rule Family.

The labels below the dotted line denote condition columns that do not serve as conclusion columns in another Rule Family.

The (Pnumber) denotes Rule Pattern numbers within the Rule Family. Where the (Px) appears with no header but a symbol [...] that indicates a pattern with no conditions.

All labels below the Rule Family name denote condition column headings.

Policy Renewal Method

Policy Tier

Within Bounds (P2, P3)

Policy Renewal Override (P1)

Policy Renewal Override

Insured Major Ownership Change (P2)

Insured Major Location Change (P1)

Policy Annual Premium (P3)

Policy Discontinued Agent (P4)

Policy Manual Flag (P5)

(von Halle & Goldberg 2010, p. 28)
Decision-aware Business Processes
Decision-aware Business Processes

- A decision-aware business process as one that is designed to distinguish between
  - tasks that perform work (i.e., process tasks) and
  - tasks that come to conclusions based on decision logic (decision tasks)

- Decision logic should be externalized from decision tasks and represented in a different kind of model
  - Process tasks are procedural
  - Decision logic should be declarative

- Separating business decisions from business process tasks
  - simplifies the business process model,
  - allows to manage business logic in a declarative form,
  - delivers the business logic in a form that transcends technology
Managing Decision-aware Business Processes

The general approach for dealing with knowledge work in business processes can be specialized to a procedure model for decision-aware business processes:

1. Process Elicitation
2. Decision Analysis: Identify key questions
3. Modelling
   - Process Flow
   - Decision Logic
4. Continuous Improvement
   - Business Process Management
   - Business Decision Management
Distinguishing a Procedural Task from a Declarative Decision

■ A procedural solution specifies how, in a step-by-step manner, something is to be done.
  ♦ So a business process model is a procedural solution because it prescribes a set of tasks that are carried out in a particular sequence.
  ♦ The business process model is the “How” of a unit of work.

■ A declarative solution only specifies what needs to be done, with no details as to how, in a step-by-step manner, it is to be carried out, because sequence is irrelevant to arriving at the correct result.
  ♦ A Decision Model is a declarative solution because it is a set of unordered business logic, not a set of ordered tasks.
  ♦ A Decision Model is the “What” of a special kind of unit of work.

(von Halle & Goldberg 2010, p. 67)
Procedural versus Declarative

A procedural solution specifies how, in a step by step manner, something is to be done.

A declarative solution is what needs to be done, with no details as to the methods to be used (no sequential information).

Business process is a procedural solution of tasks to be performed in precise sequential order. The “How” of a unit of work.

A declarative solution occurs when sequence is irrelevant to the result. The “What” of a unit of work.

(von Halle & Goldberg 2010, p. 67)
Example 1: Declarative vs. Procedural Solutions

Option 1

Start → Person Employment History → Good → Person Debt → Low → Set Person Credit Rating to A → End

Start → Person Debt → Bad → Person Employment History → High → Set Person Credit Rating to A → End

Option 2

Start → Person Debt → Low → Person Employment History → Good → Set Person Credit Rating to A → End

Start → Person Debt → High → Person Employment History → Bad → Set Person Credit Rating to A → End

Option 3

Start → Determine Person Credit Rating → End

Process Model

Rule Family Table:

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Person Debt</th>
<th>Person Employment History</th>
<th>Conclusion (Person Credit Rating)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>is Low</td>
<td>is Good</td>
<td>= &quot;A&quot;</td>
</tr>
<tr>
<td>1</td>
<td>is Low</td>
<td>is Bad</td>
<td>= (?)</td>
</tr>
<tr>
<td>1</td>
<td>is High</td>
<td>is Good</td>
<td>= (?)</td>
</tr>
<tr>
<td>1</td>
<td>is High</td>
<td>is Bad</td>
<td>= (?)</td>
</tr>
</tbody>
</table>

Decision Model Diagram

(von Halle & Goldberg 2010, p. 69)

Procedural

Declarative
Advantages of Separating Business Processes and Business Logic in Option 3

■ The Rule Family implies no particular sequence among the conditions to be tested.

■ The Rule Family indicates via the “?” that there are other possible combinations of conditions to consider.

■ The Rule Family can contain as many rows as are needed to reach the correct conclusion. It can contain additional columns if other conditions are needed to determine a person’s credit rating.

■ The Rule Family table also contains business logic for the logic not modeled in the business process models of Option 1 and Option 2, e.g. the possible values of person’s debt ("high", "low") and employment history ("good", "bad")

(von Halle & Goldberg 2010, p. 68f)
## Distinctions between Business Process and Business Decision

<table>
<thead>
<tr>
<th>Business Process</th>
<th>Business Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Procedural in nature</td>
<td>• Declarative in nature</td>
</tr>
<tr>
<td>• Consists of tasks connected by sequence</td>
<td>• Consists of Rule Families connected by inferential relationships (all independent of sequence)</td>
</tr>
<tr>
<td>• Is all about how (step-by-step sequence to carry out work)</td>
<td>• Is all about what is to be concluded (the logic leading from conditions to conclusion)</td>
</tr>
<tr>
<td>• Improvements in business process aim for increased work efficiency</td>
<td>• Improvements in a business decision aim for smarter business logic</td>
</tr>
<tr>
<td>• Represented best in a procedural business process model</td>
<td>• Represented best in a declarative Decision Model</td>
</tr>
</tbody>
</table>

*(von Halle & Goldberg 2010, p.70)*
Improvements by separating business logic from Business Process Model

- Allows a much simpler business process model
- Easily highlights all possible combinations of conditions
- Supports the principle of separation of concerns
  - Permits changes in the Decision Model without changing the business process model
  - Permits changes in the business process model without changing the Decision Model

(von Halle & Goldberg 2010, p. 69)
Disadvantages to Burying Decisions (Business Logic) in Business Processes

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Forces unnecessary sequence and constraints on business logic</td>
</tr>
<tr>
<td>2</td>
<td>Makes changes to business process and business logic difficult</td>
</tr>
<tr>
<td>3</td>
<td>Adds meaningless complexity to business logic and business process</td>
</tr>
<tr>
<td>4</td>
<td>Fails to deliver a visual representation of all business logic</td>
</tr>
<tr>
<td>5</td>
<td>Makes governance of business process and business logic difficult to manage</td>
</tr>
<tr>
<td>6</td>
<td>Results in business logic and business processes that are not reusable</td>
</tr>
<tr>
<td>7</td>
<td>Compromises SOA</td>
</tr>
</tbody>
</table>

- If a business process is too complicated, a reason might be that business rules are embedded in the flow.

- "If you separate the business rules, you can develop remarkably stable business processes" (Roger Burlton)  
  (von Halle & Goldberg 2010, p. 70)
Example 2: Business Logic not contained in a Process Model

(von Halle & Goldberg 2010, p. 71)
A Business Process Model does not Reveal All Business Logic

- If the separation of business processes and business logic is not made consciously, some business logic might be in the process model while others is missing.
- It must then be modeled separately, e.g. in the task descriptions or externally (if it is represented at all).
- Reusability if hampered: Some of the business logic may be used in several of the tasks (maybe even several processes).
- The Decision Model resurrects all of the business logic in one visual artifact.

(von Halle & Goldberg 2010, p. 72)
Managing Business Logic separately

- This solution has two tasks with their Decision Models.
- The Decision Model can be viewed, managed, and executed as one whole set of business logic, as a black box evaluating conditions and reaching a conclusion.
- Business Logic can be reused
  - the whole decision model
  - Individual rule families

(von Halle & Goldberg 2010, p. 71f)
Business Process Model, Decision Model and Rules Families

(von Halle & Goldberg 2010, p. 76)
Integrating the Decision Model with BPMN

- Execution of a decision described by a Decision Model is a particular type of task in BPMN.
- In BPMN 2.0 the corresponding task type is called a business rule task.
- In the figure below, Validate Order is the decision task. Its logic is described by the Decision Model. The gateway simply tests the output of the decision and routes the flow either to A or B based on the result.

(Von Halle & Goldberg 2010, p. 425)
**Decision Requiring Sequence**

- There are business circumstances that require separate business decisions and Decision Models.
  - Different business decision may be governed by different groups, hence having separate Decision Models facilitates separate governing bodies for the business logic.
  - There may be different decisions depending on a previous decision (Example: After a make-or-buy decision either the supplier has to be selected or the effort for the development is derived).

*(von Halle & Goldberg 2010, p. 77)*
Achieving Business Excellence by Managing Decision Logic Separately

- von Halle and Goldberg argue that operational excellence alone is insufficient for sustainable competitive advantage.

- Key business process must not only be efficient and consumer-friendly but also smart and agile
  - Business processes become agile when declarative business decisions are separated from procedural business process tasks
  - Business processes become smart when the business decisions are governed appropriately by business leaders

- When the business leadership clearly understands the business logic behind the business decisions, the impact of those decisions can be ascertained, and the business can quickly and easily make adjustments. (von Halle & Goldberg 2010, p. 78)
Literatur
