Enriching Business Process Models with Decision Rules *

Semra Catalkaya, David Knuplesch, Carolina Chiao, and Manfred Reichert

Institute of Database and Information Systems, Ulm University, Germany
{semra.catalkaya,david.knuplesch,carolina.chiao,manfred.reichert}@
uni-ulm.de

Abstract. Making the right decisions in time is one of the key tasks in every business. In this context, decision theory fosters decision-making based on well-defined decision rules. The latter evaluate a given set of input parameters and utilize evidenced data in order to determine an optimal alternative out of a given set of choices. In particular, decision rules are relevant in the context business processes as well. Contemporary process modeling languages, however, have not incorporated decision theory yet, but mainly consider rather simple, guard-based decisions that refer to process-relevant data. To remedy this drawback, this paper introduces an approach that allows embedding decision problems in business process models and applying decision rules to deal with them. As a major benefit, it becomes possible to automatically determine optimal execution paths during run time.

Keywords: business process modeling, decision support, KPI

1 Introduction

Making the right decisions during the execution of business processes is crucial for any company to achieve its business objectives [1, 2]. When choosing the wrong alternatives and execution paths during process execution, in turn, unnecessary costs or other disadvantages might result.

For instance, consider the make-or-buy process in Fig. 1 as an example of a decision-making process. Regarding this process, a manager must decide whether to produce goods locally or to outsource the production. In order to make the right decision, strategic information is required, like, for example, the time required to produce (and potentially to deliver) the goods or the production costs. However, as can be observed in the context of this example, respective information is usually not made explicit, i.e., it does not become transparent at the process model level. Hence, the manager must either query this information outside the scope of the respective process-aware information system (e.g., in an external database or application) or rely on his personal expertise.

* This work was done within the research project C^3Pro funded by the German Research Foundation (DFG), Project number: RE 1402/2-1.

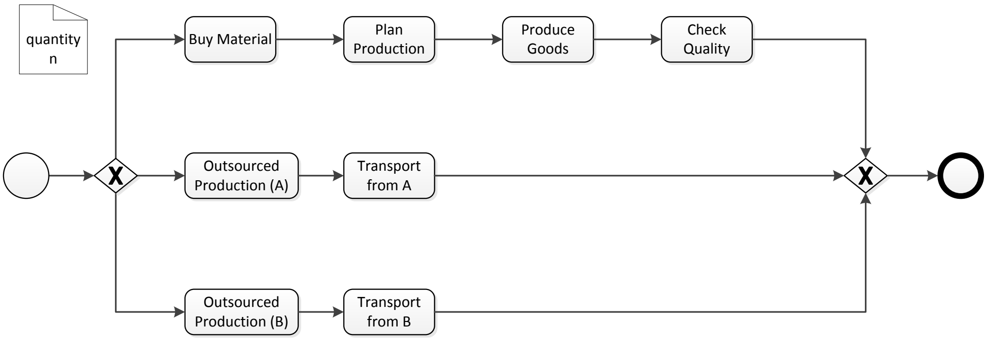

Fig. 1. Make-or-Buy Process

In management science, *decision theory* deals with the identification of factors relevant for solving particular decision problems [3]. Thereby, a *decision problem* corresponds to a problem the decision maker (i.e., organization) faces and for which decision theory is intended to provide an answer or relevant information [4]. In particular, to solve a decision problem, *decision rules* may be applied. In turn, a decision rule corresponds to a mathematical function that, given a set of parameters (e.g., production costs, time, and logistics costs), computes the best alternative based on the criteria important for a particular decision. Therefore, integrating such decision rules into business process models will make decisions in the course of a process more explicit and allow process participants to select the best alternative at a certain decision-making point.

When adding decision rules to business process models, a number of requirements must be met. First, it becomes necessary to identify the parameters that must be taken into account for a particular decision-making point (i.e., the parameters of the respective decision rule). Second, the values of these parameters have to be determined. Third, these values must be up-to-date; i.e., they must reflect the current situation of the organization, e.g., the production costs must be in accordance with current material costs. Fourth, a method is required for embedding decision rules into process models. In turn, the use of this method should be intuitive for process participants, which means that embedding and visualizing these rules within a process model must not be a complex task.

There exist very few approaches that have already applied decision theory or process mining for optimizing decision-making in the context of business processes [5–7]. However, none of them meets all of the aforementioned requirements. In general, these approaches neither provide a generic method for embedding decision rules into business process models nor do they allow for dynamic updates of parameter values required for evaluating decision rules.

The approach presented in this paper aims at overcoming these drawbacks and provides a comprehensive method for embedding decision rules into business process models. For this purpose, we introduce a modified XOR-split gateway, which we denote as *r-gateway*. The latter allows process modelers to define decision problems and decision rules directly in process models. In particular, to each r-gateway a specific decision rule may be assigned.

To be able to decide which decision-making parameters must be taken into account in the context of a particular decision rule, we utilize *key performance indicators* (KPI) as basis for our approach; e.g., costs of performing an activity. Generally, KPIs are quantifiable measurements used by an organization to analyze the success of a particular business process or business process activity. For example, such measurements may be obtained by using business intelligence (BI) tools. Furthermore, they are well aligned with the current status of the organization. Therefore, instead of manually setting values for decision-making parameters, we utilize the up-to-date KPI values obtained by BI tools.

Moreover, it is important to take into account that a branch representing a particular alternative in a process model may comprise various sets of activities. In our approach KPI values are associated with a single activity, but not with an entire execution branch of a process model. Therefore, a decision rule associated with a r-gateway computes the best alternative by aggregating all KPI values corresponding to the activities of a branch. The approach presented in this paper constitutes work-in-progress. In particular, a proper evaluation, including experimental results and a deeper investigation of potential drawbacks of our approach will be done in future work.

The remainder of this paper is structured as follows. In Section 2, fundamentals of decision theory are presented, which are required for a better understanding of this paper. Our approach is discussed in Section 3. First, an architecture of a process-aware information system is presented outlining the system components required in the context of our work. Second, we introduce the *(decision) rule-based XOR-split gateway (r-gateway)*, which allows process modelers to explicitly embed decision rules in business process models. Third, the formal semantics of the r-gateway is described. In Section 4, we illustrate the use of our approach along a simple example. Related work is discussed in Section 5. Then, Section 6 concludes the paper with a summary and an outlook.

2 Backgrounds on Decision Theory

In decision theory, three kinds of decision problems are distinguished, depending on the available knowledge [8, 9]:

- decision problems under certainty
- decision problems under risk
- decision problems under uncertainty

Generally, decision theory offers a wide range of decision rules that allow determining the best alternative, while taking different viewpoints (e.g., optimistic and risk-averse decision making). In the following, we summarize basic characteristics of decision problems under certainty, uncertainty and risk. Furthermore, we provide examples of decision rules (see [10] for a detailed description of these decision rules).

Decision problem under certainty. Regarding decision making, this problem presumes the presence of alternatives with deterministic effects [8, 11]. The *Lexicographical Order* is an examples of a decision rule addressing decision problems under certainty [11]. It chooses an alternative by considering the most important KPI for decision-making [12]. If two alternatives are equal with respect to the primary KPI, the second most important KPI will be considered as well, and so forth.

Decision problem under risk. A decision problem under risk is characterized by knowledge related to the potential effects of applying the available alternatives and to the probabilities of these effects. As opposed to a *decision problem under certainty*, the alternatives have nondeterministic behavior. The *Bayes Rule* constitutes an example of a decision rule that may be applied to decision problems under risk [9, 13]. It chooses the alternative with maximum expected value of the KPIs of the different execution branches [14].

Decision problem under uncertainty. A decision problem under uncertainty is characterized by knowledge related to the potential effects of applying the available alternatives. As opposed to a *decision problem under risk*, the probabilities of these effects are unknown[15, 16]. Decision rules like *MaxiMax*, *MaxiMin*, *Hurwicz's* and *Savage-Niehans* are options to solve decision problems under uncertainty. Regarding decision-making, the *MaxiMax* Rule chooses the alternative with the maximal best-case value of a particular KPI [17]. In turn, the *MaxiMin* Rule chooses the alternative with the maximal worst-case value of a particular KPI [15]. The *Hurwicz's Rule* combines MaxiMax and MiniMax. More precisely, for each alternative it sums up the weighted values of the worst and best case with respect to a particular KPI. Then, the alternative with the maximum sum is chosen [18]. The *Savage-Niehans Rule* (also called *MiniMax Regret Rule*) needs two steps for decision making. First, it calculates the regret values for all cases. Thereby, a regret value corresponds to the difference between the maximum value of a particular KPI for respective consequence and the value of a particular alternative for the same consequence. Following this, the rule chooses the alternative showing the smallest maximum regret value [19].

3 Embedding Decision Rules in Process Models

This section presents our approach for capturing and configuring decision problems within business process models. First, we describe components of a process-aware information system relevant in our context. Second, we show how to specify decision rules in business process models. For this purpose, we introduce a XOR-split gateway, which is associated with decision rules. Furthermore, we show how to configure such a gateway with a particular decision rule. Finally, we define the semantics of the particular XOR-split gateway at run-time.

3.1 Architecture

To enable the application of decision rules in a process-aware information system (PAIS), the latter must be able to access and process knowledge concerning the alternatives that exist in the context of a particular decision problem. However, the architecture of a PAIS enabling the use of decision rules, comprises standard components, like a process model editor, process engine, and repository (e.g., for storing activities, process models, process instances, and event logs). In our context, the process model editor must be extended to embed decision rules into process models. In turn, the BI tool analyzes process logs and stores statistical data and run-time information related to actual process execution in a respective repository (e.g., KPIs of activities, probabilities of events), i.e., *process statistics*. Finally, the process execution engine is enriched with a plug-in component evaluating decision rules at run-time. Fig. 2 outlines this architecture for decision rule support in PAIS.

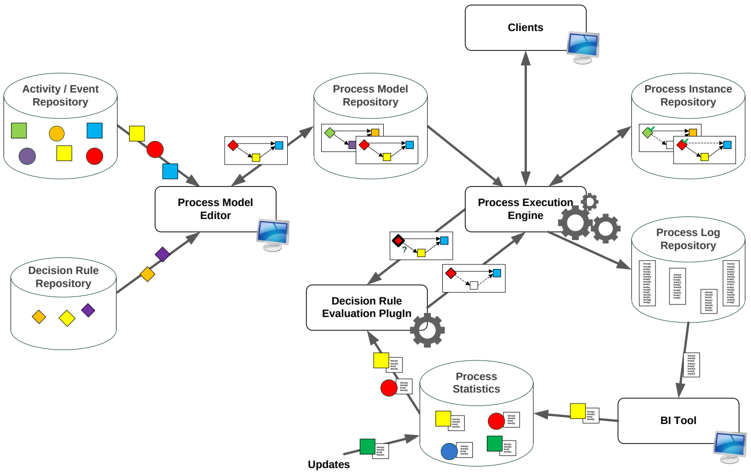

Fig. 2. Architecture of a PAIS integrating decision rules

3.2 Embedding Decision Rules in Process Models: The r-Gateway

This section introduces our approach for capturing decision problems in business process models. More precisely, respective decision problems must be annotated with a corresponding decision rule within the process model. For this purpose, we introduce the (decision) rule-based XOR-split gateway (cf. Fig. 3).

Fig. 3 illustrates how the make-or-buy decision problem from Fig. 1 can be modeled by the use of a rule-based XOR-split gateway (r-gateway). Similar to common XOR-split gateways, the r-gateway allows expressing when a decision must be made. The *configuration artifact* defines the specific decision-rule, which first computes the various KPIs of the existing alternatives, and then determines

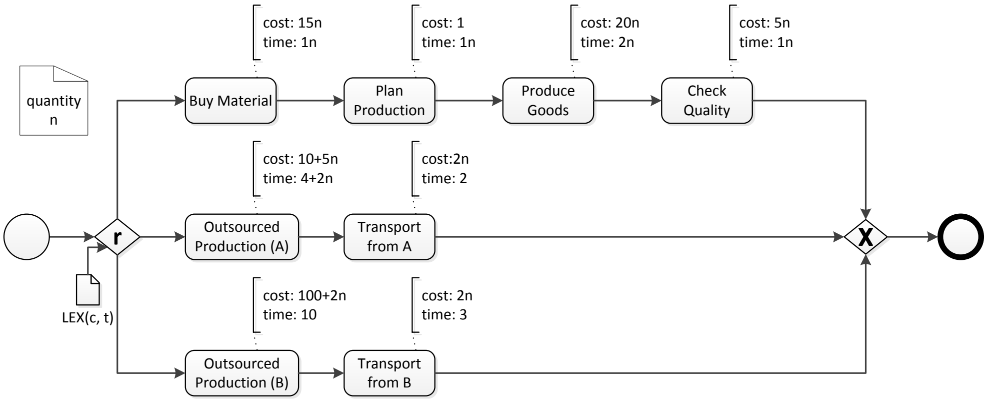

Fig. 3. Make-or-Buy Process

the optimal alternative. In our example, the configuration of the r-gateway refers to the Lexicographical Order (LEX). Further, Fig. 3 highlights the KPI values of the different activities. Note that these annotations can be added automatically (cf. Sect. 3.1). A more precise description of Fig. 3 is presented in Section 4.

3.3 Formal Framework

This section provides a set-based formalization of process models. For the sake of simplicity and to set a focus, we only consider well-structured process models in this paper. Since well-structured process models can be always represented as a *process structure tree* (PST)[20], our formalization is based on the latter (cf. Def. 2).

Process Domain. First of all, we formally introduce the notion of *process domain* (cf. Def. 1). It defines the main building blocks for modeling business processes (e.g. process fragment types, activities, events, and data objects). A process domain further comprises functions for describing relations between activities, KPIs and data objects on one hand and for accessing probabilities of events and KPI values, stored in the process statistic repository (cf. Sect. 3.1), on the other. Finally, the process domain provides functions Z^s and Z^+ to allow for the computation of KPI values of alternatives composed out of multiple activities and nested gateways.

Definition 1 (Process Domains).
A process domain $\mathfrak{D}$ *is a tuple* $\mathfrak{D} = (\mathcal{T}, \mathcal{A}, \mathcal{E}, \mathcal{O}, \mathcal{K}, chg, p, val, Z^s, Z^+, \epsilon_F)$ *with*

- $\mathcal{T} := \{s, +, d\times, e\times, r\times, dr, er, A\}$ *the set of process fragment types, whereby* s *corresponds to a sequence,* $+$ *to a parallel gateway,* $d\times$ *to a data-based xor-split,* $e\times$ *to an event-based xor-split,* $r\times$ *to a rule-based xor-split,* dr *to a data-based repetition,* er *to an event-based repetition, and* A *to an activity execution.*
- $\mathcal{A}$ *being the set of activities*
- $\mathcal{E}$ *being the set of events,*
- $\mathcal{O}$ *being the set of data objects,*

- $\mathcal{K}$ being the set of key performance indicators (KPI),
- $chg : \mathcal{O} \rightarrow 2^{\mathcal{A}}$ assigning to each data object the set of activities, that may change its value.
- $p : \mathcal{E} \rightarrow [0, 1] \cup \{undef\}$ assigning to each event a probability or $undef$ if the latter is unknown,
- $val : (\mathcal{A} \times \mathcal{K} \times \mathbb{R}^{\mathcal{O}}) \rightarrow \mathbb{R}$, assigning to each activity the expected value of a given KPI that can depend on the current state of data objects. The set of data objects that really have impact on the KPI of an activity are denoted by the function $imp : (\mathcal{A} \times \mathcal{K}) \rightarrow 2^{\mathcal{O}}$
- $Z^s : \mathcal{K} \rightarrow (\mathbb{R}^+ \rightarrow \mathbb{R})$ assigning to each KPI a function for its aggregation over fragments or activities arranged in sequence (e.g., for KPI duration Z^s might return a function that computes the sum of the parameters passed),
- $Z^+ : \mathcal{K} \rightarrow (\mathbb{R}^+ \rightarrow \mathbb{R})$ assigning to each KPI a function for its aggregation over fragments or activities arranged in parallel (e.g., for KPI duration Z^+ might return a function that computes the maximum of the parameters passed),
- ϵ_F the empty fragment.

Process Model. Based on process domain, we now introduce the notion of process model using the process structure tree [20]. According to Def. 2, a process model is composed out of fragments of different types and properties:

Definition 2 (Process model).
Let $\mathfrak{D} = (\mathcal{T}, \mathcal{A}, \mathcal{E}, \mathcal{O}, \mathcal{K}, chg, p, val, Z^s, Z^+, \epsilon_F)$ be a process domain. Any well-structured process model can then be represented as a tuple $P = (F, \lessdot, rf, type, class, evt, grd, cnf, f_r)$ with

- F being the set of fragments and f_r the root fragment.
- $\lessdot \subseteq F \times F$ being the sub-fragment relation, i.e. $g \lessdot f$ means that g is a direct sub-fragment of fragment f. Thus, $(F, \lessdot)$ is a tree with root node f_r. Based on $\lessdot$ we define
  - $\lessdot^{*} \subseteq F \times F$ as the transitive closure of $\lessdot$.
  - $|.| : F \rightarrow \mathbb{N} : f \mapsto |f| := |\{f' | f' \lessdot f\}|$ assigning to each fragment the number of its direct sub-fragments
- $type : F \rightarrow \mathcal{T}$ assigning to each fragment its type (cf. Def. 1). Based on such a type, for each $\circ \in \mathcal{T}$, we can define the set $F^{\circ} := \{f \in F | type(f) = \circ\}$ and $F^{\lessdot\circ} := \{f \in F | \exists g \in F : f \lessdot g \wedge type(g) = \circ\}$, thereby $\bigcup_{\circ \in \mathcal{T}} F^{\circ} = f_r \cup \bigcup_{\circ \in \mathcal{T}} F^{\lessdot\circ}$ are partitions of the fragment set F.
- $rf : F^{dr} \cup F^{er} \rightarrow F$ assigning to each repetition the repeated fragment, i.e. $rf(f) \lessdot f$.
- $grd : F^{d\times} \cup F^{dr} \cup F^{dr} \rightarrow \mathfrak{G}$ assigning a guard to each branch/fragment of a data-based xor-fragment. Further, grd assigns a guard to each data-based repetition fragment and the corresponding repeated fragment. Thereby, $\mathfrak{G}$ corresponds to the set of guards and a particular guard $g \in \mathfrak{G} := (\mathbb{R}^{\mathcal{O}}) \rightarrow \mathbb{B}$ constitutes a function. The latter depends on the values of data objects and decides whether or not a branch is chosen. Thereby, a guard is a function that depends on the values of the data objects and returns either `true` or `false`.
- $class : F^{A} \rightarrow \mathcal{A}$ assigning to each activity fragment an activity class.
- $evt : F^{\lessdot e\times} \cup F^{\lessdot er} \cup F^{er} \rightarrow E$ assigning an event to each branch/fragment of an event-based xor-fragment. Further, evt assigns an event to each event-based repetition fragment and the corresponding repeated fragment.
- $cnf : F^{r\times} \rightarrow \mathfrak{R}$ configuring each rule-based xor-split with a particular decision rule. Thereby, $\mathfrak{R}$ corresponds to the set of decision rules. A particular decision rule $r \in \mathfrak{R} := \mathbb{R}^{\mathcal{O}} \times (2^F - \{\emptyset\}) \rightarrow F$ constitutes a function that selects one fragment out of a set of fragments based on KPIs of the fragments. Since some KPI depend on the values of data objects, a decision rule may depend on the values of the data objects as well.

Further, we define

- $uses_O : \mathfrak{G} \rightarrow 2^{\mathcal{O}}$ that assigns to each guard the set of data objects used by the guard.
- $uses_K : \mathfrak{R} \rightarrow 2^{\mathcal{K}}$ that assigns to each decision rule the set of KPIs used by the rule.

Decision Problems. As described in Sect. 2, decision theory distinguishes between decision making under certainty, uncertainty, and risk. Accordingly, we define formal criteria for certainty, uncertainty, and risk:

Definition 3 (Criteria for Decision Problems).
Let $\mathfrak{D} = (\mathcal{T}, \mathcal{A}, \mathcal{E}, \mathcal{O}, \mathcal{K}, chg, p, val, Z^s, Z^+, \epsilon_F)$ be a process domain and $P = (F, \lessdot, rf, type, class, evt, grd, cnf, f_r)$ be a process structure tree, then:

- $f^c \in F^{r\times}$ *defines a decision problem under certainty,*
  iff $\forall f \lessdot^{*} f^c : type(f) \in \{s, +, d\times, A\}$
- $f^u \in F^{r\times}$ *defines a decision problem under uncertainty,*
  iff $\forall f \lessdot^{*} f^u : type(f) \in \{s, +, d\times, e\times, A\}$.[1]
- $f^r \in F^{r\times}$ *defines a decision problem under risk,*
  iff $\forall f \lessdot^{*} f^r : type(f) \in \{s, +, d\times, e\times, er, A\}$ *and probabilities of the used events are known,*
  i.e. $\forall f \lessdot^{*} f^r : f \in (F^{\lessdot e\times} \cup F^{\lessdot er} \cup F^{er}) \Rightarrow p(evt(f)) \neq undef$

Furthermore, we require that the value of a particular data object $o \in \mathcal{O}$ *must not be changed within a decision problem (i.e. rule-based xor-split fragment)* $f_r \in F^{r\times}$,

- *if* o *impacts the decision of a data-based xor-split sub-fragment, i.e.:*
  $\exists f_{d\times} \in F^{d\times}, f_{d\times} \lessdot^{*} f_r : o \in uses_O(grd(f_{d\times}))$
  $\Rightarrow \forall f_a \in F^A, f_a \lessdot^{*} f_r : type(f_a) \notin chg(o)$
- *or if they impact KPIs that are used by the respective decision rule* $r = cnf(f_r)$, *i.e.:*
  $\exists k \in uses_K(r) \exists f_a, \in F^A, f_a \lessdot^{*} f_r : o \in imp(type(f_a), k)$
  $\Rightarrow \forall f_{a'} \in F^A, f_{a'} \lessdot^{*} f_r : type(f_{a'}) \notin chg(o)$

3.4 Execution Semantics of the r-Gateway

This section describes the application and evaluation of decision rules at runtime. Thereby, the application of decision rules requires KPI values for each alternative. Note that in the context of business process execution, alternatives may be composed of multiple activities and nested gateways. Thus, before applying the decision rules, we first discuss how to compute KPI values of the different alternatives. Then, we formally define the application of decision rules. Due to space limitations, we only consider decision problems under risk in this paper. In [10], we additionally address decision problems under certainty and uncertainty.

Computing KPI values of alternatives under risk. In this case, the probabilities of the events are known. Furthermore, only fragments of types $s, +, d\times, e\times, er$, and A may occur (cf. Def. 3). Thus, function vfr, which computes the expected KPI values of alternatives under risk, is composed out of six subfunctions:

$vfr : F^{s,+,d\times,e\times,A} \times \mathcal{K} \times \mathbb{R}^{\mathcal{O}} \to \mathbb{R} : (f, k, a) \mapsto vfc(f, k, a) := vfr_{type(f)}(f, k, a)$ with

$$\begin{aligned}
&vfr_A(f, k, a) := val(class(f), k, a)\\
&vfr_s(f, k, a) := Z^s(\{vfr(f', k, a) | f' \lessdot f\})\\
&vfr_+(f, k, a) := Z^+(\{vfr(f', k, a) | f' \lessdot f\})\\
&vfr_{d\times}(f, k, a) := Z^+(\{vfr(f', k, a) | f' \lessdot f \wedge grd(f')(a)\})\\
&vfr_{e\times}(f, k, a) := \textstyle\sum_{f' \lessdot f} p(evt(f')) \cdot vfr(f', k, a)\\
&vfr_{er}(f, k, a) := \tfrac{1}{p(evt(f))} \cdot vfr(rf(f), k, a)
\end{aligned}$$

Application of Decision Rules. In this section, we formalize the application of decision rules during business process execution.

[1] In the context of uncertainty, the probability of each event is assumed to be unknown. We do not consider this as a prerequisite, but just ignore possibly known probabilities.

Decision rules under risk: Bayes Rule. Using function vfr, this rule determines the alternative with the maximum expected value of a particular KPI [14].

$BY : \mathcal{K} \times \mathbb{R}^{O} \times 2^{F} \to F : (\lambda, k, a, F') \mapsto BY(k, a, F')$

$BY(k, a, F') := f,$ with $\forall f' \in F^{‘} : vfr(f, k, a) \geq vfr(f, k, a)$

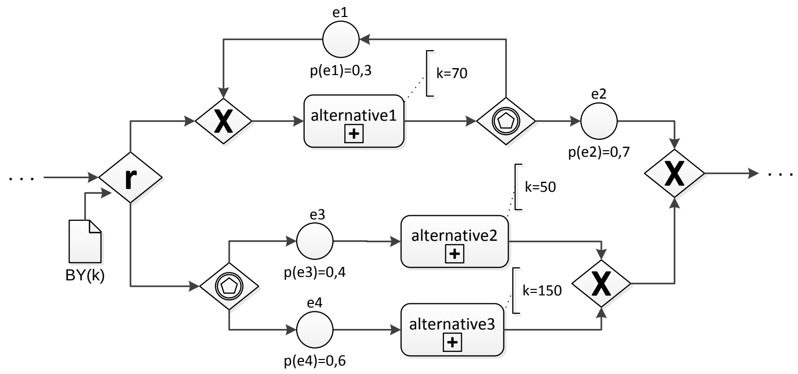

Fig. 4. Modelling of the Bayes Rule

Fig. 4 shows an example of the Bayes rule that comprises two alternatives. The first one contains an event-based loop, the second one leads to an event-based XOR-split. Based on the definition of vfr, the Bayes rule computes the expected value for both alternatives and then chooses the one with the highest expected value.

$\mathfrak{D} = (\mathcal{T}, \mathcal{A}, \mathcal{E}, \mathcal{O}, \mathcal{K}, chg, p, val, Z^{s}, Z^{+}, \epsilon_F)$ with
$\mathcal{T} := \{s, +, d\times, e\times, r\times, dr, er, \}$
$\mathcal{A} := \{a1, a2, a3\}$; $\mathcal{E} := \{e1, e2, e3, e4\}$;
$\mathcal{O} := \emptyset$; $\mathcal{K} := \{k\}$; $chg(o) := \emptyset$;

$p(e1) := 0.3; p(e2) := 0.7; p(e3) := 0.4;$
$p(e4) := 0.6; val(a1, k, 0) := 70;$
$val(a2, k, 0) := 50; val(a3, k, 0) := 150;$
$Z^{s}(k) := \sum$; and $Z^{+}(k) := \sum$;

$P = (F, \lessdot, rf, type, class, evt, grd, cnf, f_{r\times})$ with
$F := \{f_{r\times}, f_{er}, f_{e\times}, f_{a1}, f_{a2}, f_{a3}\};$
$f_{a1} \lessdot f_{er}; f_{a_2}, f_{a3} \lessdot f_{e\times}; f_{er}, f_{e\times} \lessdot f_{r\times};$
$rf(f_{e\times}) := f_{a1}; grd := false;$
$type(f_{r\times}) := r\times; type(f_{er}) := er;$
$type(f_{e\times}) := e\times; type(f_{a1}) := A;$

$type(f_{a2}) := A; type(f_{a3}) := A;$
$class(f_{a1}) := a1; class(f_{a2}) := a2;$
$class(f_{a3}) := a3;\ evt(f_{er}) := e2; evt(f_{a1}) := e1;$
$evt(f_{a2}) := e3; evt(f_{a3}) := e4;$
$cnf(f_{r\times}) := BY(k)$; and $uses_{K}(BY(k)) := \{k\}$;

Note that $f_{r\times}$ denotes a decision problem under risk:
$\forall f \lessdot^{*} f_{r\times} : type(f) \in \{s, +, d\times, e\times, er, A\}$ and function p never returns $undef$.

$vfr(f_{a1}, k, 0) = val(class(f_{a1}), k, 0) = val(a1, k, 0) = 70;$

$vfr(f_{a2}, k, 0) = 50; vfr(f_{a3}, k, 0) = 150;$

$vfr(f_{er}, k, a) := \frac{1}{prop(evt(f_{er}))} \cdot vfr(rf(f_{er}), k, 0) = \frac{1}{prop(e2)} \cdot vfr(f_{a1}, k, 0) = \frac{1}{0.7} \cdot 70 = 100;$

$vfr(f_{e\times}, k, 0) = \sum_{f' \lessdot f_{e\times}} prop(evt(f')) \cdot vfr(f', k, 0)$

$= prop(evt(f_{a2})) \cdot vfr(f_{a2}, k, 0) + prop(evt(f_{a3})) \cdot vfr(f_{a3}, k, 0) = 0.4 \cdot 50 + 0.6 \cdot 150 = 110;$

$\Rightarrow BY(k, 0, \{f_{er}, f_{e\times}\}) = f_{e\times},$ because $vfr(f_{e\times}, k, 0) \geq vfr(f_{er}, k, 0)$, i.e., the second alternative $(f_{e\times})$ will be chosen.

4 Application Example

This section applies our approach to the example from Fig. 1. In this context note that Fig. 3 models the same make-or-by process. However, the latter comprises the rule-based XOR-split gateway introduced in Section 3.2. In this process, the decision maker must decide whether to produce goods locally or to outsource the production. In case of outsourcing, the goods may be produced by two different suppliers. Depending on the quantity of the ordered goods, in turn, the suppliers may be characterized by different production costs and delivery times. In this example, the *Lexicographical Order* rule determines the optimal alternative based on the KPI related to production costs. If two alternatives have the same KPI value, the second important KPI value is considered (e.g., delivery time).

Regarding our example from Fig. 1, the company established a demand of $n = 1500$ pieces. With the use of the aforementioned rule, three alternatives must be considered: The first alternative is the local production of goods, which requires a total time of $t = 5 \cdot 1500 = 7500$ days and consumes total costs of $c = 1 + 40 \cdot 1500 = 60001$ €. The second alternative is divided into two steps and requires a total time of $t = 6 + 2 \cdot 1500 = 3006$ days and costs $c = 10 + 7 \cdot 1500 = 10510$ €. In turn, the third alternative has duration $t = 10{+}3 = 13$ days and costs $c = 100 + 4 \cdot 1500 = 6100$ €.
Finally, as result of applying the Lexicographic Rule, we learn that the third alternative shall be taken. Of course, this simple example does not validate our approach, but just shall illustrate how the r-gateway works. A detailed evaluation is subject of future work.

5 Related Work

There exist several approaches that provide operational decision support in the context of process-aware information system (PAIS). For instance, [21] and [22] describe a simulation system for operational decision support in the context of workflow management. The approach combines and extends the workflow system YAWL and the process mining framework ProM. However, respective approaches do not offer methods for decision-making at run-time. In addition, they are not able to dynamically update the KPI values according to the current status of the company. The approaches presented in [23] and [24] provide operational decision support in terms of recommendations at run-time. However, these approaches do not provide a method for embedding decision rules in business process models. In our approach, we do not explicitly address the discovering of KPIs. In the context of process mining [6] and business intelligence (BI) [25], there exist approaches tackling this challenge. [26] provides a framework for performance monitoring and analysis of WS-BPEL processes, which consolidates process events and KPI measurements. Additionally, [26] analyzes dependencies between KPIs by applying machine learning techniques. The approach described in [5] offers a toolbox of decision rules and methods to enable and ease the development of decision support systems.

Compared to this related work, our approach allows for the explicit specification of decision problems in business processes models. The use of KPIs as parameters for decision rules ensures that decisions will be made based on up-to-date values. Finally, it enables automated decision-making at run-time.

6 Conclusion

In this paper, we presented an approach for applying decision theory techniques to process models. For this purpose, we first outlined the adaptations to be made regarding the architecture of traditional PAIS. Then, we introduced the *(decision) rule-based XOR-split gateway (r-gateway)* to enrich current standards and notations for process modeling. The latter indicates the occurrence of a decision problem and must be configured with a particular decision rule. This rule shall be executed at run-time when enabling the respective r-gateway. To support an automated evaluation of decision rules, we formally described how to compute KPI values of alternatives composed out of multiple activities and nested gateways. Thereby, KPI values of single activities are discovered from process logs through the application of business intelligence tools.

Note that this paper only serves as a starting point for embedding decision theory concepts in business process management and process execution. Thus, our next steps will include a proof-of-concept implementation as well as a proper evaluation, including experimental results and the identification of potential drawbacks. Furthermore, we plan to consider properties of KPIs that depend on random distributions.

In general, the applicability of our approach will be highly dependent on the availability of accurate KPIs for each activity. Thereby, another challenge is to adapt current business intelligence technologies for optimizing our approach. Finally, additional challenges emerge in the context of adaptive and flexible processes [27, 28], e.g., regarding the dynamic addition of new activities and process fragments that might be relevant for decision making.

References

1. Lohrmann, M., Reichert, M.: Efficacy-aware business process modeling. In: CoopIS'12. LNCS (2012) 38–55
2. Lohrmann, M., Reichert, M.: Modeling business objectives for business process management. In: S-BPM ONE'12. (2012) 106–126
3. Einhorn, H.J., Hogarth, R.M.: Behavioral decision theory: Processes of judgment and choice. Journal of Accounting Research **19**(1) (1981) 1–31
4. Newell, A., Simon, H.A.: Human problem solving. Volume 14. Prentice-Hall (1972)
5. Worley, J.H., Castillo, G.R., Geneste, L., Grabot, B.: Adding decision support to workflow systems by reusable standard software components. Computers in Industry **49**(1) (2002) 123–140
6. van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer (2011)

7. Rozinat, A., Mans, R., van der Aalst, W.M.P.: Mining cpn models: discovering process models with data from event logs. In: CPN'06, Citeseer (2006)
8. Joyce, J.M.: The foundations of causal decision theory. Cambridge University Press (1999)
9. Peterson, M.: An introduction to decision theory. Cambridge University Press (1993)
10. Catalkaya, S., Knuplesch, D., Reichert, M.: Bringing more semantics to xor-split gateways in business process models based on decision rules. Technical Report 2013-04, University of Ulm (2013)
11. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica **47** (1979) 263–291
12. Peeta, S., Yu, J.W.: A hybrid model for driver route choice incorporating en-route attributes and real-time information effects. Networks and Spatial Economics **5**(1) (2005) 21–40
13. Wu, G., Zhang, J., Gonzalez, R.: Decision under risk. In: Blackwell handbook of judgment and decision making. Blackwell (2004) 399–423
14. Pawlak, Z.: Decision rules, bayes rule and rough sets. In: New directions in rough sets, data mining, and granular-soft computing. Springer (1999) 1–9
15. Kartam, N.A., Tzeng, G.H., Teng, J.Y.: Robust contingency plans for transportation investment planning. IEEE Transactions on Systems, Man, and Cybernetics **23**(1) (1993) 5–13
16. Dixit, A.: Entry and exit decisions under uncertainty. Journal of political Economy **97** (1989) 620–638
17. Arnold, B.F., Größl, I., Stahlecker, P.: The minimax, the minimin, and the hurwicz adjustment principle. Theory and decision **52**(3) (2002) 233–260
18. Haimes, Y.Y., Lambert, J.H., Li, D.: Risk of extreme events in a multiobjective framework. Journal of the American Water Resources Association **28**(1) (1992) 201–209
19. Rödder, W., Reucher, E.: A consensual peer-based dea-model with optimized cross-efficiencies–input allocation instead of radial reduction. European Journal of Operational Research **212**(1) (2011) 148–154
20. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. Data & Knowledge Engineering **68**(9) (2009) 793–818
21. Rozinat, A., et al.: Workflow simulation for operational decision support. Data & Knowledge Engineering **68**(9) (2009) 834–850
22. de Medeiros, A.K.A., van der Aalst, W.M.P., Pedrinaci, C.: Semantic process mining tools: core building blocks. In: ECIS'08. (2008)
23. Schonenberg, H., et al.: Supporting flexible processes through recommendations based on history. In: BPM'08. Springer (2008) 51–66
24. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Product based workflow support: dynamic workflow execution. In: CAiSE'08, Springer (2008) 571–574
25. zur Mühlen, M., Shapiro, R.: Business process analytics. In: Handbook on Business Process Management 2. Springer (2010) 137–157
26. Wetzstein, B., et al.: Monitoring and analyzing influential factors of business process performance. In: EDOC'09, IEEE (2009) 141–150
27. Reichert, M., Dadam, P.: ADEPT$_{flex}$ – supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems, Special Issue on Workflow Management Systems **10**(2) (1998) 93–129
28. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer (2012)