

Modeling Business Objectives for Business Process Management

Matthias Lohrmann and Manfred Reichert

Ulm University,
Databases and Information Systems Institute,
{matthias.lohrmann,manfred.reichert}@uni-ulm.de

Abstract. For application scenarios such as the management of business process variants or business process quality, business objective models assume the role of formal requirements definitions as in software engineering. However, effective concepts in this area still constitute a gap in the presently available array of business process management methods. To address this issue, this paper develops and shortly evaluates a refined business objective modeling approach. Our approach builds on use case-based effectiveness criteria, and on insights gained from assessing the state of the art. It derives required constructs and interrelations from application scenarios, and integrates these into a business objective meta-model. As an initial validation of our concept, we model a sample scenario and match the results against effectiveness criteria.

Key words: Business Objectives, Business Goals, Business Process Modeling, Business Process Quality

1 Introduction

Definitions for the term *business process* range from business process reengineering approaches [1, 2] to trade associations [3] and current business process management (BPM) research [4]. They generally comprise a notion of *business goals* or *business objectives*, which can be viewed as an integral aspect reflecting the utilitarian nature of *business* processes. It is therefore interesting that formal modeling of business objectives is not covered by common business process modeling approaches such as BPMN [5], EPCs [6] or Workflow Nets [7].

To further illustrate the potential benefits of concise business objective modeling, we consider its possible use cases. In general, business objective models could assume the role of a formal requirements definition for business process design, implementation and enactment. Accordingly, application scenarios and benefits are generally comparable to broader requirements engineering [8, 9].

From an a priori perspective, business objective models add a layer of abstraction to business process models. They allow discussing and documenting business objectives independently from a concrete implementation, and assessing the effectiveness of implementation options. From an ex post perspective, they can be used to determine whether a business process instance (or a set

of instances) has terminated in a state being consistent to the organization’s business targets. In addition to these general use cases, we point out three more specific exemplary scenarios that are closely related to present directions of BPM research: automated derivation of business process models, management of business process variants, and business process quality management.

Scenario 1 (Automated Derivation of Core Business Process Models). An available business objective model would comprise formalized information on the things a respective business process should create or alter including decision criteria, ordering constraints, etc. On that basis, it would be possible to formally derive minimum control flow implementation requirements. In addition, it would be possible to integrate the approach with process mining techniques [10]. For instance, real-world cases might be used to analyze the probability of control flow decision criteria given in a business objective. This would allow for automated or even continuous control flow optimization.

Scenario 2 (Management of Business Process Variants). The management of business process variants has emerged as an important BPM issue [11]. However, determining whether two process instances are variants of the same business process remains a “missing link” in this respect, especially when considering the mining of process variants or reference processes [12], or the refactoring of model repositories [13]. Formally modeling business objectives could contribute to closing this gap, as it would enable to assess the “equivalence” of process variants with respect to a business objective.

Scenario 3 (Business Process Quality Management). Business process quality management constitutes another emerging area of BPM research. In [14], a definition framework in this regard was developed. The concept of *efficacy*, i.e., whether a business process achieves its business objective, is crucial in this respect. Thus, formal modeling of business objectives constitutes an important prerequisite to effectively assess business process quality in design, implementation and enactment.

Scenario 4 (Subject-oriented Business Process Management). Subject-oriented business process management (S-BPM) constitutes an approach to address critical practical issues in BPM adoption, thereby significantly broadening its appeal [15]. The concept is based on shifting the paradigm of BPM away from formalizing tasks to be executed to the roles and interactions of subjects or stakeholders. As this potentially takes out some of the implicit formality of, e.g., strict control flow, it is all the more important to employ concise business objective models to ensure effectiveness, i.e. making sure that a process still achieves what it should.

Our research has shown that present approaches to business objective modeling do not yet effectively support the application scenarios set out above. This paper therefore seeks to develop a refined solution which is methodologically well-founded as well as readily applicable in future research and for practical purposes. The remainder of the paper is structured as follows: Section 2 presents our methodology including process examples to illustrate our ideas, basic terminology, and effectiveness criteria to evaluate results. Sections 3, 4, and 5 implement our methodology, from a review of available approaches to a refined business objectives meta-model. Section 7 concludes the paper and gives an outlook on future research.

2 Methodology and Preliminary Considerations

A business objective model constitutes a goal-bound artificial construct. We therefore apply the principles of design science [16, 17]. Accordingly, our methodology consists of the following steps:

1. Define *effectiveness criteria* to assess the utility of design artifacts in the field of business objectives modeling (cf. Section 2.3).

2. Assess the *state of the art* based on the defined effectiveness criteria to determine gaps and obtain pointers towards a refined solution (cf. Section 3).
3. *Build* required terminology for business objectives based on effectiveness criteria and research into available approaches (cf. Section 4).
4. *Build* a meta-model for business objectives (cf. Section 5).
5. *Evaluate* our solution with respect to effectiveness criteria (cf. Section 6).
6. Discuss implications and further steps to leverage our results (cf. Section 7).

The remainder of this section presents two sample business processes we use to illustrate and evaluate our results, discusses required preliminary terms, and develops effectiveness criteria to evaluate present approaches as well as our work.

2.1 Sample Processes

Our first sample process stems from the field of accounting: *invoice checking and approval* constitutes a typical example of an administrative process which is often supported by workflow management systems.¹

To ensure that our concepts are also applicable to domains where workflow applications are not as common, we include a second sample process from the field of healthcare. Figures 1 and 2 show the models of the two sample processes in terms of BPMN [5] flow charts.

Example 1 (Sample Process A: Invoice Checking and Approval). The business process starts with the receipt of a supplier invoice (activity A1). The invoice is then compared to the respective purchase order (A2). If deviations exist, these are subject to approval. In practice, this is often the case when, for instance, price data have not been maintained or when no purchase order has been entered into the ERP system. If the deviation is approved (A3), the purchase order is created or adapted (A4). Otherwise, the invoice is declined (A6, A7). In the next step, the invoice is matched against goods receipt (A5) and, depending on the result, either declined (A6, A7) or passed to the next check, which is based on the invoice value. For a value of more than 5,000, senior management approval (A8) is required. If this is granted, the invoice may finally be approved (A9).

Example 2 (Sample Process B: Medical Examinations). In our alternate example from the healthcare field, a medical examination A is performed (B1). Based on its result, a drug is applied (B2) and a second examination B (B3) is performed or not. A third examination C, which may only be carried out once examination A is completed, should follow in each case (B4). Thereafter, another drug is applied depending on the result of examination C (B5) and the age of the patient. In parallel, further steps are performed depending on the results of examinations A and B: First, the existence or non-existence of condition X is noted dependent on the result of examination C (B6, B7). Then, a fourth examination D is performed (B8). After completing examination D, application of a drug is required (B9).

2.2 Basic Terminology

Business processes constitute artifacts in the sense of design science [16] which operate within an *affecting and affected outer environment*. The outer environment of a business process consists of *target artifacts* and *resources*, i.e. things

¹ Incoming invoices processing was used to illustrate the concept of business process reengineering by both Davenport and Hammer [1, 2].

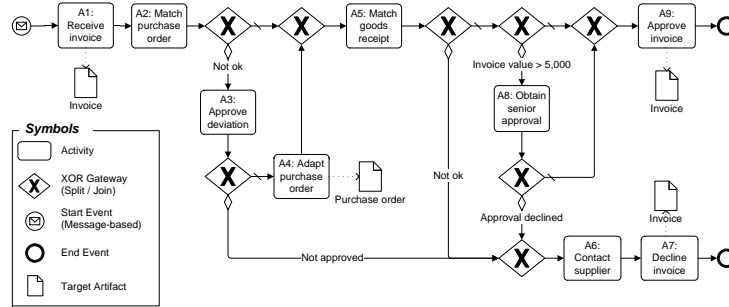


Fig. 1. Sample Process A: Invoice checking and approval

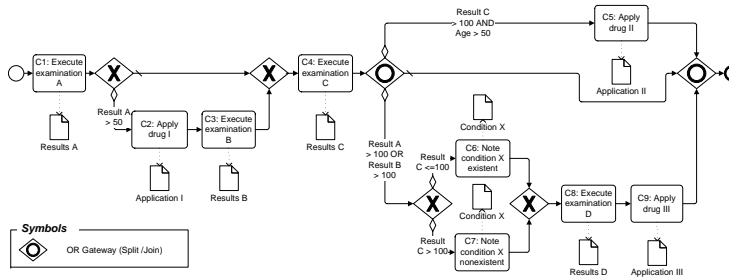


Fig. 2. Sample Process B: Medical examinations

the process strives to create and alter, and things required to properly do so. Note that this perspective differs in some regards from the classic BPM concepts of process input and process output as it includes things usually not considered (e.g. capital goods). This topic was discussed in more detail in [14].

In the field of BPM, business objectives represent the targets an organization aims to achieve with a business process. As illustrated in Example 3, this can be understood on a strategic, collective operational or transactional level.

Example 3 (Semantic Business Objective Levels). As another exemplary business process, consider the handling of job applications in an enterprise. On a *strategic level*, the business objective of this process may be understood as providing the organization with the right “human resources”. On a *collective operational level*, the business objective may be understood as properly handling the overall occurring cases of job applications. Depending on the required service level, the business objective may, for instance, be fulfilled if 90% of cases are managed correctly. On a *transactional level*, it may be understood as properly handling an individual application.

For the purpose of business objectives modeling, we define the term business objective on the transactional level to achieve consistency with common business process modeling approaches: In business process modeling, models are generally defined on a process instance [3] level without considering the cardinality of cases or instances. This means that a task that occurs many times for the business process, but one time per process instance is modeled as an individual activity, not as a set of activities.

Moreover, remember that an affecting environment may determine what actually needs to be induced to fulfil a business objective, for instance when considering decision processes (cf. Example 4).

Example 4 (The Affecting Environment of Business Objectives). Again, consider the job application process from Example 3. In this case, the business objective cannot be achieved by simply approving or disapproving an application. Rather, the respective hiring criteria are to be considered. Thus, they constitute the affecting environment of the business objective. As another example, consider medical treatments. In many cases, tests are required to find out which drugs are required. In this case, the test results are part of the affecting environment of the business objective.

Note that, when considering business objective levels as well as the affecting environment, the *organizational target* as the business objective on strategic level may differ from business objectives on lower semantic levels. This occurs when the affecting environment restrains the business process from achieving the original organizational target (cf. Example 5). In other words, there may be states of the affecting environment where the business objective of the process is fulfilled while the corresponding organizational target has not been achieved.

Example 5 (Business Objective Levels and the Affecting Environment). When handling incoming job applications, the strategic organizational target will be to fill the respective positions. However, the business objective on more operational levels may well pertain to decline an applicant if her qualifications (as part of the affecting environment) are not sufficient.

In summary, this leads us to the following basic definition for business objectives to be further elaborated in our modeling approach:

Definition 1 (Business Objective). *A **business objective** in the sense of business process management constitutes a refinement of organizational targets to the transactional level. It pertains to an affecting and affected environment. The affecting and affected environment represent the things to be considered and the things to be manipulated to achieve the business objective. The business objective relates each state of its relevant affecting environment to a set of aspired states of the affected environment.* □

2.3 Effectiveness Criteria

Considering the scenarios lined out in Section 1, business objective models as requirements definitions for business processes will generally be used to

- determine what needs to be done to achieve a business objective (e.g., as a starting point for structured business process design, or as in Scenario 1 from Section 1),
- assess whether a modeled business process enables to achieve its business objective (e.g., to evaluate design options, or as in Scenario 2), and
- assess whether a concrete business process instance has actually achieved its business objective (e.g., in testing, or as in Scenario 3).

Accordingly, the notion of an *achieved* function reflecting whether an aspired state of the affecting and affected environment of a business objective is reached is central to business objectives modeling.

Recapitulating the terms introduced in Section 2.2, business objectives are *achieved* by propagating target artifacts to an *aspired state*. However, which

<i>SR1</i>	<i>Consideration of the affecting environment:</i> Whether a business objective is achieved or not must be determined in terms of target artifacts <i>and</i> additional properties of the outer environment; e.g., in Sample Process A (cf. Example 1), the approved or disapproved invoice as a target artifact and the defined conditions for invoice approval as additional properties of the outer environment.
<i>SR2</i>	<i>Varying target environment:</i> The set of target artifacts to be created or altered as well as the concrete operations to be carried out on them may vary; e.g., in Sample Process A (cf. Example 1), the purchase order may have to be adapted or not, but the invoice must always be approved or disapproved.
<i>SR3</i>	<i>Order constraints:</i> There may be constraints regarding the order in which the activities of a process need to be executed in conformance with the business objective. Consider, for instance, Sample Process B from Example 2: drug application and examinations must occur in a specific order. It is important to note that these constraints actually represent constraints with respect to target artifacts manipulation, because, by definition, executing activities cannot constitute a business objective.
<i>UC1</i>	<i>Semantic interdependencies:</i> The approach should be apt to transparently capture semantic interdependencies between elements of the outer environment like mutual exclusivity or correlation. As an example for mutual exclusivity, consider the approval or disapproval of invoices in Sample Process A from Example 1 (cf. “pragmatic quality” in the sense of comprehension in [18]).
<i>UC2</i>	<i>Model compaction:</i> The approach should lead to a compact result in the sense of avoiding unnecessary content which might “hide” the relevance of model elements. For instance, in Sample Process A, it would be obstructive to model the effect of senior management approval for invoices below a value of 5,000 (cf. “semantic quality” in the sense of validity or relevance to the problem in [18]).
<i>UC3</i>	<i>Knowledge externalisation:</i> The approach should leverage implicitly available knowledge of the modeler (cf. “physical quality” in the sense of externalisation in [18]).

Table 1. Effectiveness Criteria for Business Objective Modeling Approaches

target artifacts need to be created or altered, and which states are considered as aspired may depend on other elements of the affecting environment.² Thus, business objectives cannot be recorded solely in terms of attributes of targets artifacts, but in terms of a set of consistency rules to be satisfied in respect to the entire environment. This set of rules must be complete and free of overlaps to ensure conformance can be assessed for each state of the outer environment.

Table 1 summarizes *effectiveness criteria* towards business objectives modeling. The *semantic requirements* SR1 to SR3 are based on the issues discussed above. They reflect the semantic content an approach needs to address to properly model business objectives. In addition, an effective modeling approach will also fulfill *usability criteria* UC1 to UC3 to support both modelers and users. The usability criteria are based on the considerations on model quality in [18]. Since we work on a meta-model level instead of the model level addressed in [18], we place special regard to the quality types of “physical quality”, “semantic quality” and “pragmatic quality”.

² Note that the affecting environment of a business objective may differ from the affecting environment of an associated business process – the affecting environment of an efficacious business process will encompass, but possibly not be limited to, the affecting environment of its business objective (cf. [14]).

3 State of the Art

Models for business objectives or goals³ have been proposed by Kueng and Kawalek [19], Neiger and Churilov [20], Soffer and Wand [21], and Lin and Sølvsberg [22]. Markovic and Kowalkiewicz [23] proposed a business goal ontology as part of the SUPER project on semantic BPM (cf., e.g., [24]). For comparison, we include an approach by Engelman et al. towards goals modeling in enterprise architecture. Table 2 matches the approaches against semantic requirements SR1 to SR3. For reasons of brevity, usability criteria UC1 to UC3 are not considered.

Source / focus	Evaluation against semantic requirements (cf. Table 1)		
	SR1	SR2	SR3
<i>Kueng and Kawalek</i> [19]: Goals-based modeling, design evaluation	<i>Not fulfilled:</i> No formal measurable definition of goals	<i>Not fulfilled:</i> Goals are discussed on an abstract level only	<i>Not fulfilled:</i> Goals are discussed on an abstract level only
<i>Neiger and Churilov</i> [20]: “Value-focused thinking” to structure objectives	<i>Not fulfilled:</i> No formal measurable definition of objectives	<i>Not fulfilled:</i> “Functional objectives” on a more abstract level	<i>Not fulfilled:</i> “Functional objectives” on a more abstract level
<i>Soffer and Wand</i> [21]: Formalizing processes’ contribution to “soft goals”	<i>Not fulfilled:</i> Business goals as any possible process termination state, goal achievement only pertains to target artifacts	<i>Partially fulfilled:</i> Implicitly considered: only one relevant process path required per target artifact	<i>Partially fulfilled:</i> Order constraints implicitly considered via consistent process paths
<i>Lin and Sølvsberg</i> [22]: Goal ontology for semantic annotation in distributed environments	<i>Partially fulfilled:</i> Goals are seen as states of activities or artifacts, but no specification of respective artifact states	<i>Not fulfilled:</i> Goals are defined for activities instead of processes, no concept of goals changing with the environment	<i>Partially fulfilled:</i> Constraints are comprised in the meta-model, but not further specified as state of activities or the environment
<i>Markovic and Kowalkiewicz</i> [23]: Integrating goals into business process modeling	<i>Not fulfilled:</i> No concise definition of when a goal has been achieved	<i>Not fulfilled:</i> No notion of goals evolving with the environment	<i>Not fulfilled:</i> No notion of order constraints
<i>Engelman et al.</i> [25]: Enterprise architecture goals modeling language	<i>Not fulfilled:</i> Hard goals concept, but no formal notion of goal achievement	<i>Not fulfilled:</i> No affecting environment concept	<i>Partially fulfilled:</i> Goal aggregation might be extended to include ordering

Table 2. Available Business Objective Modeling Approaches

Since the related approaches discussed generally aim at amending process models with a descriptive goals perspective and not necessarily at using business objectives as a formal requirements definition in a BPM context, it is not

³ In the field of BPM, the terms are generally used as synonyms.

surprising that additional work is needed to develop a *business objectives meta-model* to fully address the criteria set out in Table 1.

4 Extended Business Objective Modeling Terminology

According to semantic requirement SR1 in Table 1, an effective approach to business objectives modeling must relate aspired states of target artifact properties to conditional states of additional properties of the outer environment. In the following, we will refer to the respective environmental properties as *elements of the target environment* (or, in short, *target elements*) and *elements of the conditional environment* (or, in short, *conditional elements*). Both sets of *environmental elements* may overlap, i.e., an environmental element may constitute a target element, a conditional element or both. We may conceive of environmental elements as “metering points” of sufficient semantic relevance to determine, in their totality, whether a business objective has been *achieved*. Note that the conditional elements correspond to the additional properties of the outer environment cited in semantic requirement SR1 in Table 1. The relevant “metering points” may be expressed as *binary state determinants*.

Definition 2 (Binary State Determinants). *A **binary state determinant (BSD)** is the combination of an environmental element with an absolute or relative state condition that is relevant to a business objective and that may or may not be **fulfilled**. **Conditional BSDs** and **target BSDs** refer to conditional elements and a target elements, respectively.*

On that basis, it would be possible to list all BSDs with respect to conditional elements, enumerate the possible states and relate them to the corresponding set of aspired states of the target elements, represented by target BSDs. This approach would link aspired target states to the affecting environment as demanded by semantic requirement SR1 (cf. Table 1). However, there would still be major issues regarding our effectiveness criteria, as presented in Table 3.

To address these topics, we introduce a business objectives modeling approach that (i) reflects distinct types of target BSDs, (ii) sets out with target BSDs instead of conditional BSDs, and (iii) avoids redundancies in its modeling of both the target and the conditional environment. To this end, we employ a number of terms summarized in the remainder of this section.

Definition 3 (Target BSD types). ***Target BSDs** are constituents of the business objective. To achieve a business objective, all respective target BSDs must assume **target values**. Dependent on the range of target values, we discern various target BSD types.*

*To achieve the business objective, **monovalent target BSDs** must assume a “true” value (target BSDs that may only assume a “false” value are to be rephrased accordingly). There is no condition attached. Note that target BSDs subject to order constraints must include “false” in their value range.*

<i>SR2</i>	As all target BSDs are enumerated for each conditional state, the potentially limited relevance of individual target artifacts is “hidden”.
<i>SR3</i>	Order constraints are not addressed, and still require an additional construct.
<i>UC1</i>	Semantic interrelations between elements of the outer environment, such as mutual exclusivity or correlation, are not captured.
<i>UC2</i>	For an individual target BSD, only a (typically small) part of the conditional environment is relevant. Hence, a relation matrix between conditional and target BSDs would only be sparsely populated. For instance, in Sample Process B (cf. Figure 2), the age of the patient is not relevant to examination B. This characteristic is not utilized which leads to a unnecessarily bloated model.
<i>UC3</i>	From a modeler’s perspective, it is much easier to determine (e.g. by discussion with stakeholders) what the prerequisite conditions for a target BSD are than which target BSDs are determined by a conditional element, let alone <i>a priori</i> enumerating relevant conditional BSDs. Moreover, semantic interrelations or relevances (cf. UC1-2) are not addressed. Capturing available knowledge is thus impeded.

Table 3. Basic Modeling vs. Effectiveness Criteria

To achieve the business objective, **fully determinate bivalent target BSDs** may assume either a “true” or a “false” value. We thus require only one condition attached to either “true” or “false”.

To achieve the business objective, **partially determinate bivalent target BSDs** may assume either a “true” or a “don’t care” value (“false” target BSDs are to be rephrased).⁴ “True” is bound to a respective condition.

To achieve the business objective, **trivalent target BSDs** may assume a “true”, a “false”, or a “don’t care” value. Trivalent target BSDs differ from bivalent ones as there are two conditions attached to “true” and “false”. The conditions are mutually exclusive, but not comprehensive (i.e. one or none of the two can evaluate to “true” at the same time). □

Table 4 provides an overview on the various target BSD types and the state they must assume to enable achieving the business objective depending on the state of their relevant conditional environment.

Note that trivalent target BSDs can also be understood as two partially determinate bivalent target BSDs referring to the same target element. However, modeling a trivalent target BSD as two bivalent target BSDs results in a loss of semantics because the two respective bivalent target BSDs’ mutual exclusivity is not visible in the model.

Definition 4 (Conditional propositions). *Conditions attached to target BSDs can be expressed as **conditional propositions** consisting of conjunctively and / or disjunctively interlinked **conditional BSDs**. Unlike target BSDs, the value*

⁴ “Don’t care” implies that the business process needs to do nothing – consider, for instance, the target BSD “Purchase order value = invoice value” from Sample Process A in Figure 1, where we either need to adapt the purchase order or simply leave it as it is. Semantically, this represents the characteristic that the set of relevant target artifacts may change with the conditional environment.

Target BSD types	Condition states	Aspired target BSD states		Example
		Not fulfilled	Fulfilled	
<i>Monovalent</i>	n/a		X	Examination A in Sample Process B
<i>Fully determinate bivalent</i>	Not fulfilled Fulfilled	X	X	Invoice approval in Sample Process A
<i>Partially determinate bivalent</i>	Not fulfilled Fulfilled	X	X X	Senior management approval in Sample Process A
<i>Trivalent</i>	Only 1st condition fulfilled Only 2nd condition fulfilled No condition fulfilled Both conditions fulfilled	X X X May not occur	X X X	Marking of condition X in Sample Process B

Table 4. Target BSD Types

range of conditional BSDs is confirmed to “true” and “false”. A target element may also act as a conditional element within one business objective.

Absolute conditional BSDs compare one conditional element to an absolute value range. **Relative conditional BSDs** compare two conditional elements to each other.

Target BSDs are considered as **conditionally equivalent** if the attached conditional propositions are equivalent or if, for fully determinate bivalent target BSDs, the attached conditional propositions are a negation of each other. Target BSDs are considered as **conditionally dependent** on another if a BSD’s conditional proposition comprises the value another target BSD has assumed or should assume by way of a relative conditional BSD. \square

We identified the treatment of order constraints as a requirement towards business objective modeling (see semantic requirement SR3 in Table 1). To address this issue, we consider a number of characteristics of conditional propositions as specified in Definition 4:

- As shown in Example 6, a conditionally dependent target BSD shares conditional proposition of the “father” BSD.
- A conditional dependency exists for any two target BSDs where an *order constraint* applies; i.e., the dependent target BSD must be fulfilled before, after or at the same time as the “father” BSD.
- From a modeling perspective, it does not make a difference which BSD is the “dependent” one, because both are required to achieve the business objective.

We therefore introduce a convention to model conditional dependencies and order constraints as described in Table 5.

Note that conditionally dependent target BSDs to be fulfilled *at the same time* should be merged with their “father” BSD (i.e., the two underlying target elements should be treated as one as they must be manipulated concurrently anyway) or resolved into two (or more) sequences as appropriate.⁵

Example 6 (Order Constraints Modeling). Consider Sample Process B in Figure 2. The activity pairs B2 / B3 and B8 / B9 reflect that examination B has to be prepared by applying a drug while

⁵ Note that this issue is also not addressed in common process modeling approaches.

<i>Order constraint: the conditionally dependent target BSD must be fulfilled...</i>	<i>Modeled conditional propositions</i>	
	<i>“Father” target BSD</i>	<i>Conditionally dependent target BSD</i>
... before	Dependent target BSD only	Shared conditional proposition only
... after	Shared conditional proposition only	“Father” target BSD only
... at any time (no order constraint)	Shared conditional proposition only	Shared conditional proposition only

Table 5. Order Constraints Modeling

another drug is required after examination D. The applications of both drugs thus become elements of the target environment which are conditionally dependent on the respective examination.

In the first case, application of drug I is dependent on whether examination B *shall happen*. In the second case, the application of drug II is dependent on whether examination D *has happened*. Regardless of the requirements with respect to the order of activities, both drug applications are semantically dependent on the relevant examination and thus share the examination’s conditional environment. However, they differ in terms of their order constraint in regard to the respective examination. Nevertheless, both are part of the business objective, which – given the respective conditional environment – cannot be fulfilled unless the drugs are applied properly.

The considerations set out above enable to define business objective achievement on the basis of target BSDs and conditional propositions, thus addressing semantic requirement SR1 in Table 1.

Definition 5 (Business objective achievement). *A business objective is achieved iff each target BSD comprised in the business objective has assumed a state reflecting its conditional propositions. Thus, a business process has to approve or disapprove each conditional proposition and manipulate target artifacts accordingly.* □

Based on Definition 5 and our convention on the modeling of order constraints in Table 5, control flow in business processes is generally oriented at approving and disapproving conditional propositions. Optimizing control flow thus amounts to optimizing the approval and disapproval of conditional propositions that are, in turn, composed of conditional BSDs.

Accordingly, control flow in business processes will be designed based on the business objective’s conditional propositions. It thus makes sense to refine the business objectives meta-model to represent properties of conditional propositions which are relevant to approval or disapproval. To this end, we discern necessary and sufficient sub-conditions as possible constituents of conditional propositions. In case of multiple conditional BSDs comprised in a conditional proposition, necessary and sufficient sub-conditions generally occur in pairs. In case of one conditional BSD in a conditional proposition, the conditional BSD amounts to the sole necessary and sufficient sub-condition.

Definition 6 (Necessary and sufficient sub-conditions). *For conditional proposition $CP := NC_1 \wedge NC_2$, NC_1 and NC_2 constitute **necessary sub-conditions**. Any part of a conditional proposition that is conjunctively linked to the entire remainder of the conditional proposition (e.g. any subterm in a*

conjunctive normal form) constitutes a necessary sub-condition. If any one necessary sub-condition is not fulfilled, the conditional proposition is disapproved.

For conditional proposition $CP := SC_1 \vee SC_2$, SC_1 and SC_2 constitute **sufficient sub-conditions**. Any part of a conditional proposition that is disjunctively linked to the entire remainder of the conditional proposition (e.g. any sub-term in a disjunctive normal form) constitutes a sufficient sub-condition. If any sufficient sub-condition is fulfilled, the conditional proposition is approved. \square

Sufficient and necessary sub-conditions can be identified by building minimal conjunctive and disjunctive normal forms for each conditional proposition (e.g., by way of a Karnaugh-Veitch diagram). The respective subterms provide us with minimal ways to either approve or disapprove a target BSD. As they are relevant for any business process implementation of a business objective, we include them in our semantic business objectives meta-model.

To fully capture the semantic content of business objectives either formally or based on *a priori* knowledge, we also consider semantic interrelations between target BSDs beyond conditional equivalence or dependency (cf. Definition 4). Target BSDs may be semantically correlated or mutually exclusive. In our context, semantic correlation e.g. for two target BSDs infers that if the first BSD is required to achieve the business objective, the second BSD will be required as well.⁶ Mutual exclusivity implies that the business objective cannot be fulfilled if two respective target BSDs are both fulfilled (i.e., $TargetBSD_1 \Rightarrow \neg TargetBSD_2$ and $TargetBSD_1 \Rightarrow \neg TargetBSD_1$). This is caused by “overlaps” in the conditional environment, i.e. conditional BSDs that are relevant for multiple target BSDs or in themselves correlated or mutually exclusive. Table 6 summarizes the possible semantic interrelations between two fully determinant bivalent target BSDs that occur with common sub-conditions.

Semantic interrelation	Type of common sub-condition X			
	Target BSD A		Target BSD B	
	Necessary	Sufficient	Necessary	Sufficient
Common conditional branch	X		X	
Mutually exclusive (fully determinate bivalent only)	X		\bar{X}	
B is correlated to A	X			X
[No proposition]	X			\bar{X}
Possibly common approval		X		X
[No proposition]		X		\bar{X}

Note: \bar{X} refers to an inversed sub-condition

Table 6. Semantic Target BSD Interrelations

⁶ Note that temporal concurrency would be an even more strict requirement, as it would demand that target BSDs are fulfilled *at the same time*.

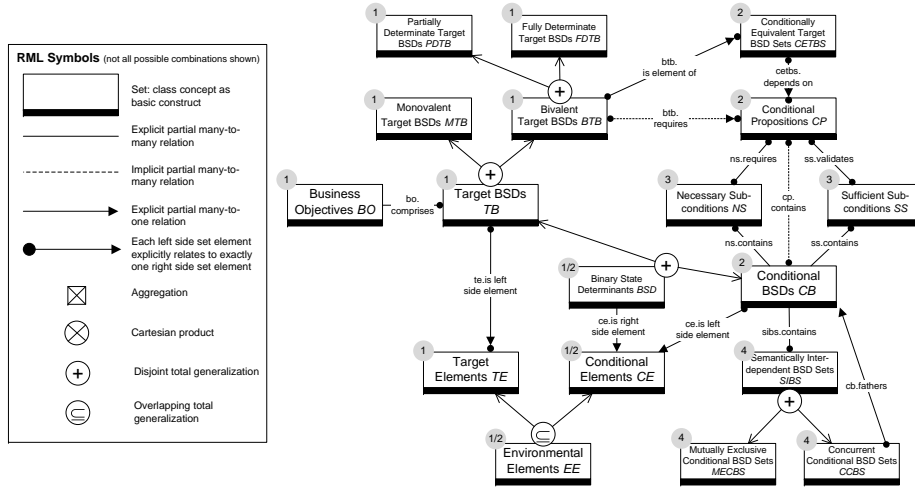


Fig. 3. Business Objective Meta-model

Besides common sub-conditions, mutual exclusivities and concurrencies may also be caused by semantic interdependencies between conditional BSDs. Beyond the simple case of non-overlapping value ranges for conditional BSDs referring to a common conditional element, it is, however, not practical to capture these characteristics in business objective modeling. Accordingly, an effective semantic business objective model will reflect multiple occurrences of individual necessary or sufficient sub-conditions in various conditional propositions linked to target BSDs as well as mutually exclusive and concurrent conditional BSDs referring to a common conditional element.

5 Business Objective Meta-model

The semantic concepts discussed in the previous section can be integrated into the RML⁷ meta-model presented in Figure 3.

The following modeling steps illustrate how a business objectives model which is compliant to the presented meta-model can be obtained. Following these steps is a possibility we suggest with regard to criterium UC3 from Table 1. The numbering included in Figure 3 reflects the order of modeling steps. Relevant explanatory notes regarding modeling concepts and their interrelations are comprised as well. Capital letters represent sets of constructs where all elements are of the same type. Section 6 will apply the respective steps to a sample process.

Step 1 (List Target BSDs). Based on the Business Objective’s target artifacts, all relevant Target BSDs including their types are listed. The respective

⁷ We use the Referent Model Language (RML) notation defined in [26] because it provides a concise graphical notation for set theory constructs.

Conditional Propositions may be modeled in a later step to make use of implicitly available knowledge on the business process first and limit modeling effort.

A Business Objective bo comprises a set of Target BSDs TB_{bo} . The Business Objective is achieved iff all comprised Target BSDs have assumed a target value.

According to Definition 3, a Target BSD might be a Monovalent Target BSD $mtb \in MTB$ or a Bivalent Target BSD $btb \in BTB$, i.e.,

$$TB = MTB \dot{\cup} BTB$$

A Bivalent Target BSD might be a Fully Determinate Target BSD $fdtb \in FDTB$ or a Partially Determinate Target BSD $pdtb \in PDTB$, i.e.,

$$BTB = FDTB \dot{\cup} PDTB$$

Note that we choose to model trivalent Target BSDs as two Partially Determinate Target BSDs as described in Section 4. Target BSDs and Conditional BSDs $cb \in CB$ are Binary State Determinants $bsd \in BSD$, i.e.,

$$BSD = TB \dot{\cup} CB$$

A Binary State Determinant bsd consists of a left-side element $leftside_{bsd}$, a right-side element $rightside_{bsd}$, and a relation rel_{bsd} , i.e.,

$$bsd = (leftside_{bsd}, rightside_{bsd}, rel_{bsd} \in \{=, <, >, \dots\})$$

Each Target BSD tb refers to a Target Element $te \in TE$, and each Conditional BSD cb to a Conditional Element $ce \in CE$ as its left-side element, i.e.,

$$leftside_{bsd} \in \begin{cases} TE & \text{if } bsd \in TB \\ CE & \text{if } bsd \in CB \end{cases}$$

Each Binary State Determinant bsd refers to a Conditional Element $ce \in CE$ or to an absolute value range $vr \in VR$ as its right-side element, i.e.,

$$rightside_{bsd} \in CE \dot{\cup} VR$$

Target Elements $te \in TE$ and Conditional Elements $ce \in CE$ are Environmental Elements $ee \in EE$. A Target Element may also be a Conditional Element, i.e.,

$$EE = TE \cup CE$$

A BSD $bsd \in BSD$ is fulfilled iff its relation rel_{bsd} holds between its left-side element $leftside_{bsd}$ and its right-side element $rightside_{bsd}$, i.e.,

$$fulfilled(bsd) := leftside_{bsd} rel_{bsd} rightside_{bsd} \quad \square$$

Step 2 (Normalize Bivalent Target BSDs). To “normalize” Bivalent Target BSDs, we build conditionally equivalent sets. To limit modeling effort, normalization can initially be conducted based on implicit knowledge without formally considering Target BSDs’ Conditional Propositions.⁸

⁸ As an example for implicit available knowledge, consider Sample Process C: a physician will know that examination B requires drug I without modeling conditions first.

According to Definition 4, Fully Determinate Target BSDs are “rephrased” (i.e. negated) to join a conditionally equivalent set if the respective Conditional Proposition is a negation of a set’s joint Conditional Proposition. Note that this is semantically not possible for Partially Determinate Target BSDs. Each Bivalent Target BSD $bsd \in BSD$ is an element of one Conditionally Equivalent Target BSD Set $cetbs_{btb}$ sharing one Conditional Proposition $cp_{cetbs_{btb}}$. Conditional Propositions are then made explicit as logical expression of Conditional BSDs considering the convention for order constraints in Table 5. A Conditional Proposition cp is fulfilled iff its logical expression is fulfilled, i.e.,

$$fulfilled(cp) := \begin{cases} true & \text{if the logical expression for } cp \text{ is fulfilled} \\ false & \text{else} \end{cases}$$

On that basis and according to Definition 5, a Business Objective bo is fulfilled iff the states of its Target BSDs and the respective Conditional Propositions are coherent considering Target BSD types, i.e.,

$$\begin{aligned} achieved(bo) := & \forall mtb \in MTB_{bo} : fulfilled(mtb) \wedge \\ & \forall fdtb \in FDTB_{bo} : fulfilled(cp_{fdtb}) \Leftrightarrow fulfilled(fdtb) \wedge \\ & \forall pdtb \in PDTB_{bo} : fulfilled(cp_{pdtb}) \Rightarrow fulfilled(pdtb) \quad \square \end{aligned}$$

Step 3 (Resolve Conditional Propositions). Conditional Propositions are resolved into Necessary and Sufficient Sub-conditions according to Definition 6. Each Conditional Proposition can be decomposed into a set of Necessary Sub-conditions NS_{cp} and a set of Sufficient Sub-conditions SS_{cp} , i.e.,

$$\begin{aligned} fulfilled(cp) & \Leftrightarrow \forall ns \in NS_{cp} : fulfilled(ns) \\ & \Leftrightarrow \exists ss \in SS_{cp} : fulfilled(ss) \end{aligned}$$

Each Necessary Sub-condition ns and each Sufficient Sub-condition ss contain a set of least one Conditional BSD CB_{ns} or CB_{ss} . A Necessary Sub-condition ns is fulfilled iff at least one of its Conditional BSDs is fulfilled, i.e.,

$$fulfilled(ns) \Leftrightarrow \exists cb \in CB_{ns} : fulfilled(cb)$$

A Sufficient Sub-condition ss is fulfilled iff all of its Conditional BSDs are fulfilled, i.e.

$$fulfilled(ss) \Leftrightarrow \forall cb \in CB_{ss} : fulfilled(cb)$$

Necessary and Sufficient Sub-conditions are modeled in consolidated form, i.e., equivalent sub-conditions for multiple Conditional Propositions are modeled only once. The decomposition of Conditional Propositions into sub-conditions can also be used to identify conditional equivalences not recognized yet. \square

Step 4 (Consolidate Conditional BSDs). To consolidate Conditional BSDs, we identify Semantically Interdependent BSD sets. A Semantically Interdependent BSD Set $sibs$ comprises a number of Conditional BSDs CB_{sibs} and may

either be a Mutually Exclusive Conditional BSD Set *mecbs* or a Concurrent Conditional BSD Set *ccbs*. Each Mutually Exclusive Conditional BSD Set comprises at least two Conditional BSDs with:

$$\text{fulfilled}(cb) \mid cb \in CB_{mecbs} \Rightarrow \nexists ecb \in (CB_{mecbs} \setminus cb) : \text{fulfilled}(ecb)$$

Each Concurrent Conditional BSD Set comprises at least one Conditional BSD and refers to one Conditional BSD $cb_{father_{ccbs}}$ which “fathers” the set:

$$\text{fulfilled}(cb_{father_{ccbs}}) \Rightarrow \forall ccb \in CB_{ccbs} : \text{fulfilled}(ccb)$$

Mutual exclusivity of Conditional BSDs propagates to Necessary Sub-conditions that consist of just the one Conditional BSD, rendering the respective Conditional Propositions and hence Target BSDs mutually exclusive as well. Semantic correlation propagates to Sufficient Sub-conditions that consist of just the one Conditional BSD, rendering the respective Conditional Propositions and hence Target BSDs semantically correlated as well.⁹

Mutual exclusivity and semantic correlation are most obvious if the respective BSD set relates to the same Conditional Element. In that case, mutual exclusivity is caused by non-overlapping value ranges, and correlation is caused by partial quantity relations in value ranges. However, this is not a strict prerequisite.

Being aware that not all interdependencies in the outer environment are generally known to the modeler, note that this modeling step will usually lead to a partial result reflecting best knowledge. \square

6 Sample Validation

This section presents an initial validation of the approach presented in Section 5 through application to the medical examination process from Figure 2 followed by a short evaluation against our effectiveness criteria.

6.1 Sample Application

We retrace the steps presented in Section 5 for Sample Process B:

Step 1 (List Target BSDs including types). For Sample Process B in Example 2, note that “Examination C executed” is not monovalent due to order restrictions (Examination C can only be executed after Examination A). Moreover, we assume that medical examinations as well as medications are not arbitrary, i.e. they should only be executed in case of a clear indication. Note that the originally trivalent Target BSD “Condition X marked” is deconstructed into two Partially Determinate Target BSDs. Results are presented in Table 7. \square

Step 2 (Normalize Bivalent Target BSDs). There are no Conditionally Equivalent Target BSD Sets containing more than one Target BSD in our example, as illustrated in Table 8. For comparison, we also show how the normalized Target BSD sets would change when not considering order constraints. \square

Target BSDs	Target BSD types
Result A available	Monovalent
Drug I applied	Fully determinate bivalent
Result B available	Fully determinate bivalent
Result C available	Fully determinate bivalent
Drug II applied	Fully determinate bivalent
Condition X marked	Partially determinate bivalent
Condition X not marked	Partially determinate bivalent
Result D available	Fully determinate bivalent
Drug III applied	Fully determinate bivalent

Table 7. Sample Target BSDs

Result with consideration of order constraints:

$CETBS_{bo}$	BSD types	Conditional Propositions
Result A available	Monovalent	none
Drug I applied	Fully determinate bivalent	[Result A > 50]
Result B available	Fully determinate bivalent	[Drug I applied]
Result C available	Fully determinate bivalent	[Result A available]
Drug II applied	Fully determinate bivalent	[Result C > 100] AND [Age > 50]
Condition X marked	Partially determinate bivalent	([Result A > 100] OR [Result B > 100]) AND [Result C ≤ 100]
Condition X not marked	Partially determinate bivalent	([Result A > 100] OR [Result B > 100]) AND [Result C > 100]
Result D available	Fully determinate bivalent	[Result A > 100] OR [Result B > 100]
Drug III applied	Fully determinate bivalent	[Result D available]

Alternative result without consideration of order constraints:

$CETBS_{bo}$	BSD types	Conditional Propositions
Result A available, Result C available	Monovalent	none
Drug I applied, Result B available	Fully determinate bivalent	[Result A > 50]
...
Result D available, Drug III applied	Fully determinate bivalent	[Result A > 100] OR [Result B > 100]

Table 8. Sample Normalization of Target BSDs

Step 3 (Resolve Conditional Propositions). Table 9 shows the resolution of Conditional Propositions into Necessary and Sufficient Sub-conditions. □

Figure 4 presents a graphical notation of the results up to now based on the exemplary content for Sample Process B. The format is simplified as it presents either Necessary or Sufficient Sub-conditions (in case of only one Conditional BSD comprised in a Conditional Proposition, the differentiation is unnecessary). Because modeling is executed in a consolidated form, there is just one “column” for each Conditional BSD or Sub-condition comprised in Figure 4. Conditional Elements which are also target Elements (this is the case for all Conditional Elements except the patient’s age) are comprised in the “line” of the respective Target BSD. The figure is to be read as follows: to achieve the Business Objective,

- the monovalent Target BSD set must be fulfilled,

⁹ See Table 6 for semantic relations caused by common sub-conditions.

$CETBS_{bo}$	Conditional Propositions	
	Necessary Sub-conditions	Sufficient Sub-conditions
Drug I applied	[Result A > 50]	[Result A > 50]
Result B available	[Drug I applied]	[Drug I applied]
Result C available	[Result A available]	[Result A available]
Drug II applied	((Result C > 100) ([Age > 50]))	((Result C > 100] AND [Age > 50])
Condition X marked	((Result A > 100] OR [Result B > 100]) ([Result C ≤ 100])	((Result A > 100] AND [Result C ≤ 100]) ([Result B > 100] AND [Result C ≤ 100])
Condition X not marked	((Result A > 100] OR [Result B > 100]) ([Result C > 100])	((Result A > 100] AND [Result C > 100]) ([Result B > 100] AND [Result C > 100])
Result D available	((Result A > 100] OR [Result B > 100])	((Result A > 100]) ([Result B > 100])
Drug III applied	[Result D available]	[Result D available]

Table 9. Sample Target BSDs with Resolved Conditional Propositions

- all elements of Bivalent Target BSD sets for which we modeled Necessary Sub-conditions must be fulfilled if all Sub-conditions for the set are fulfilled, and
- all elements of Bivalent Target BSD sets for which we modeled Sufficient Sub-conditions must be fulfilled if at least one Sub-condition for the set is fulfilled.

Note that circular relations between Target BSDs (i.e., one Target BSD as Conditional Element of another which is also a Conditional Element of the first Target BSD etc.) must not occur, because in that case the Business Objective could not be achieved by any business process. Figure 4 can thus be read from the top down.

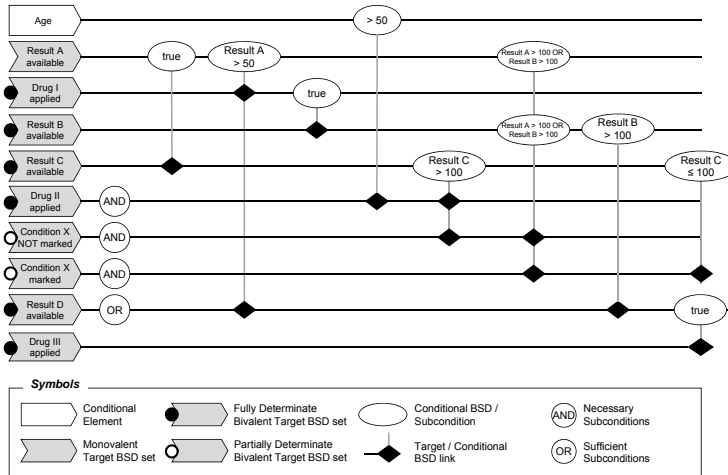


Fig. 4. Sample Conditional Consolidation

Step 4 (Consolidate Conditional BSDs). In our sample case, we only consolidate on the basis of Conditional Elements shared between Conditional BSDs, i.e., we assume no further semantic interrelations between Conditional BSDs. Consolidation results can thus be easily derived from Figure 4, as we only need to consider line by line:

- Concurrent Conditional BSD Set: [Result A > 50] \Rightarrow [Result A available]
- Mutually Exclusive Conditional BSD Set: [Result C > 100] $\Leftrightarrow \neg$ [Result C \leq 100]

Accordingly, application of Drug I and Examination C are correlated, and marking Condition X is mutually exclusive with application of Drug II and (obviously) not marking Condition X. \square

6.2 Evaluation against Effectiveness Criteria

To evaluate our results, we consider the criteria defined in Table 1:

- *SR1*: The approach builds on target and conditional elements. Accordingly, both relevant aspects of the outer environment are covered effectively.
- *SR2*: The relevance of Target BSDs is determined considering the conditional environment. Together with Partially Determinate Bivalent Target BSDs, this enables target artifact sets varying with the conditional environment.
- *SR3*: Order constraints can be modeled via a convention (cf. Table 5).
- *UC1*: Semantic interdependencies are captured via the normalization of Target BSDs and conditional consolidation. Necessary and sufficient sub-conditions can directly be used to optimize control flow via approval / disapproval strategies.
- *UC2*: The resulting model is compact and apt for graphic presentation (cf. Fig. 4). Imagine, for comparison, full enumeration of the conditional environment and the related aspired states. There are no redundant model elements.
- *UC3*: By setting out with target elements, modeling is, in our opinion, intuitive and less prone to errors of omission. The approach also allows capturing available semantic knowledge before formal modeling. Available modeler knowledge could be captured through the “guided” modeling steps – however, this topic is obviously subject to individual preferences.

7 Conclusion

We developed an approach to business objective modeling by deriving a semantically enriched meta-model and a corresponding modeling methodology. The approach fulfills semantic requirements deducted from typical application scenarios as well as additional effectiveness criteria for practical adoption. Most prominently, and in contrast to related work, it addresses both the affecting and the affected environment of business objectives. We intend future work in this area to focus on promising application scenarios facilitated by our approach to business objectives. As an example, consider automated ongoing optimization of control flow from Scenario 1. Leveraging the concept of necessary and sufficient sub-conditions might be very beneficial in this respect. Beyond the use cases lined out already, we aim at exploring additional areas of application such as formal control of business process chains in functionally structured organizations or in service-oriented architectures.

References

1. Davenport, T.J., Short, J.E.: The new industrial engineering: Information technology and business process redesign. *Sloan Mgmt. Rev.* (4) (1990) 11–27
2. Hammer, M.: Reengineering work: don't automate, obliterate. *Harvard Bus. Rev.* **68**(4) (1990) 104–112
3. WfMC: Workflow Management Coalition terminology & glossary 3.0 (1999) <http://www.wfmc.org>, document reference WFMC-TC-1011.
4. Weske, M.: *Business Process Management*. Springer (2007)
5. The Object Management Group: *Business Process Model and Notation: Version 1.2* (2009) <http://www.omg.org/spec/BPMN/1.2>.
6. Scheer, A.W., Thomas, O., Adam, O.: Process Modeling Using Event-Driven Process Chains. In: *Process-aware Information Systems*. Wiley (2005) 119–145
7. van der Aalst, W.M.P.: The application of Petri nets to workflow management. *J. Circ. Syst. Comput.* **8**(1) (1998) 21–26
8. Hull, E., Jackson, K., Dick, J.: *Requirements Engineering*. 3rd edn. Springer (2011)
9. van Lamsweerde, A.: Goal oriented requirements engineering: a guided tour. In: *Proc. 5th RE, IEEE* (2001)
10. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: a research agenda. *Comput. Ind.* **53**(3) (2004) 231–244
11. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. *J. Sw. Mnt. Ev. Res. Pract.* **22**(6-7) (2010) 519–546
12. Li, C., Reichert, M., Wombacher, A.: Mining business process variants: Challenges, scenarios, algorithms. *Data & Knowl. Eng.* **70**(5) (2011) 409–434
13. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. *Comput. Ind.* **62**(5) (2011) 467–486
14. Lohrmann, M., Reichert, M.: Understanding Business Process Quality. In: *Advances in Business Process Management*. Springer (2012) accepted for publication.
15. Fleischmann, A.: What is S-BPM? *CCIS* **85** (2010) 85–106
16. Simon, H.A.: *The Sciences of the Artificial*. 3rd edn. MIT Press (1996)
17. March, S.T., Smith, G.F.: Design and natural science research on information technology. *Decis. Support Syst.* **15**(4) (1995) 251–266
18. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. *Europ. J. of Inf. Sys.* **15** (2006) 91–102
19. Kueng, P., Kawalek, P.: Goal-based business process models: creation and evaluation. *Bus. Process Manag. J.* **3**(1) (1997) 17–38
20. Neiger, D., Churilov, L.: Goal-oriented business process modeling with EPCs and value-focused thinking. In: *Proc. 2nd BPM, LNCS 3080* (2004) 98–115
21. Soffer, P., Wand, Y.: On the notion of soft-goals in business process modeling. *Bus. Process Manag. J.* **11**(6) (2005) 663–679
22. Lin, Y., Sølvsberg, A.: Goal annotation of process models for semantic enrichment of process knowledge. In: *Proc. 19th CAiSE, LNCS 4495* (2007) 355–369
23. Markovic, I., Kowalkiewicz, M.: Linking business goals to process models in semantic business process modeling. In: *Proc. 12th EDOC, IEEE* (2008) 332–338
24. Pedrinaci, C., Domingue, J., Alves de Medeiros, A.K.: A core ontology for business process analysis. In: *Proc. 5th ESWC, LNCS 5021* (2008) 49–64
25. Engelsman, W., Quartel, D., Jonkers, H., van Sinderen, M.: Extending enterprise architecture modelling with business goals and requirements. *Ent. Inf. Sys.* **5**(1) (2011) 9–36
26. Sølvsberg, A.: Data and what they refer to. In: *Conceptual Modeling*. Springer (1999) 211–226