
Organisational reference models: supporting an adequate design of local business processes

Iris Reinhartz-Berger and Pnina Soffer*

Management Information Systems Department,
University of Haifa,
Carmel Mountain, Haifa 31905, Israel
Email: iris@mis.haifa.ac.il
Email: spnina@is.haifa.ac.il
*Corresponding author

Arnon Sturm

Department of Information Systems Engineering,
Ben-Gurion University of the Negev,
Beer Sheva 84105, Israel
Email: sturm@bgu.ac.il

Abstract: Reference models, whose aim is to capture domain knowledge, can assist in the adequate design of enterprise specific business processes. In complex organisations, business processes can be locally designed by specific organisational units. However, in order to be adequate, these processes should meet the local needs while maintaining the organisational standards. For this purpose, we propose to create an organisational reference model that specifies the organisational standards, guides and constrains the different organisational units when designing their specialised local processes. We propose a reference modelling approach called application-based domain modelling (ADOM), which is capable of specifying guidelines and constraints as part of the reference model and validating a specific model against the reference model. The paper presents the principles of ADOM and in particular, its novel validation procedure. This procedure enables the organisation to ensure that the local processes are in compliance with the organisational standards, as specified in the reference model. We demonstrate the validation procedure on a purchase requisition reference model within a university and its application in two sub units: a library and an acquisition department.

Keywords: process modelling; reference model; reuse; business process validation and verification; domain analysis; business process modelling notation; BPMN.

Reference to this paper should be made as follows: Reinhartz-Berger, I., Soffer, P. and Sturm, A. (xxxx) 'Organisational reference models: supporting an adequate design of local business processes', *Int. J. Business Process Integration and Management*, Vol. X, Nos. Y, pp.000–000.

Biographical notes: Iris Reinhartz-Berger is a Faculty Member at the MIS Department, University of Haifa. She received her PhD in 2003 in Information Management Engineering from the Technion, Israel Institute of Technology. Her research interests include conceptual modelling, modelling languages and techniques for analysis and design, domain analysis, development processes and methodologies. Her work has been published in international journals and conferences.

Pnina Soffer is a Faculty Member at the MIS Department, University of Haifa. Her PhD and MSc were in the Faculty of Industrial Engineering and Management at the Technion – Israel Institute of Technology. Her research areas are business processes and conceptual modelling, including ontological foundations of both.

Arnon Sturm is a Faculty Member at Ben-Gurion University. He received his PhD in Information Management Engineering from the Technion, Israel Institute of Technology. His research interests include conceptual modelling, modelling languages and techniques for analysis and design, domain engineering, agent-oriented software engineering and systems development processes.

1 Introduction

Reference models capture generic business knowledge, which can contribute to the adequate design of specific business processes in enterprises. While reference models have been used for this purpose for over a decade, they can also be used for meeting the specific challenges of adequate process design in complex and distributed organisations. Such organisations, which include a large number of loosely-connected organisational units, employ a variety of business processes. Yet, in order to function as a single business entity, some standardisation of the business processes is needed. This is particularly important when all the organisational units share the same enterprise system. When designing business processes in such environments, an adequate process design should balance two conflicting goals. The first goal is that the specific processes would meet the diverse needs of each unit. These needs may vary due to, e.g., localisation requirements or differences in the practices and constraints of each particular unit. At the same time, the other goal is to keep organisational standards, so that some common business logic is applied, and similarity among the processes is maintained as much as possible.

To address these conflicting goals, this paper proposes to create ‘organisational reference models’. These reference models should be as generic as possible, and provide the common business logic (that should be applied across the organisation) and the allowed variability. They can then be specialised and customised to the specific needs of the different organisational units. Finally, the specific process models can be validated against the organisational reference models, to verify that the business processes are adequately designed in compliance with the required standards and constraints.

The importance of validating the specific designed business processes is threefold. First, it maintains standardisation of the processes from the business point of view. Second, it may be required for compliance of all the organisational units with external standards and legislature. Third, implementing and maintaining the running business processes is much easier when these all have a solid common infrastructure, thus it is desired that sub units are aligned with that infrastructure.

Reference models, which are models used for supporting the construction of other models (Thomas, 2005), have been addressed for over a decade. Traditionally, reference models have not been used within an organisational context but rather as generic models whose aim is to provide generic knowledge and assist in business process design in specific enterprises. The focus of attention has mostly been the construction of reference models and the knowledge that is captured in them and recently also the process of reusing this knowledge for constructing specific processes (Mendling et al., 2005; Recker et al., 2005; Rosemann and Aalst, 2007). Validation of the specific processes to assess their compliance with the reference model has not been, to the best of our knowledge, explicitly proposed so far.

The proposed approach relies on a well-established discipline of domain engineering (Czarnecki and Eisenecker, 2000; Gomaa, 2005; Kang et al., 1990, 1998; Moon et al., 2005) and adopts the application-based domain modelling (ADOM) approach (Reinhartz-Berger and Strum, 2007; Sturm and Reinhartz-Berger, 2004) to reference models. ADOM is proposed as a platform for creating organisational reference models, applying them when constructing specific models according to the needs of an organisational unit and validating the latter against their corresponding reference models. ADOM is based on a three layered architecture: application, domain and language. The domain layer consists of specifications of various application families (domains), while the application layer consists of particular systems or business processes. The language layer includes meta-models of (modelling) languages. ADOM enforces constraints among the different layers. Particularly, the domain layer enforces constraints on the application layer, while the language layer enforces constraints on both the application and domain layers. These constraints provide support to both the construction of models at the application layer and their validation.

When adopting ADOM to reference models, the reference models reside in the domain layer, specifying and enforcing constraints on the application models, which are the particular business processes in various units of the same organisation or consortium. Thus, an approach that provides guidelines and validation templates when utilising reference models is established. ADOM is a generic approach and can be applied to different modelling languages. In Reinhartz-Berger and Strum (2007) and Sturm and Reinhartz-Berger (2004), for example, ADOM is applied to UML and in Soffer et al. (2007) – to EPC. In this paper we use ADOM with the business process modelling notation (BPMN) (OMG-BMI, 2006) due to its evolvment as a standard and its expressiveness. BPMN belongs to a recent generation of process modelling languages which has evolved based on experience gained with various business process modelling languages, such as UML activity diagram and EPC. A specification in BPMN is based on a flowcharting technique tailored for creating graphical models of business process operations (OMG-BMI, 2006). Furthermore, BPMN models can be created using CASE tools, such as Eclipse (2007) and Intalio (2007), and can be transformed to BPEL specifications (Ouyang et al., 2006), making them particularly suitable for large organisations that wish to employ service-oriented architecture (SOA) in their IT infrastructure.

We refer to the utilisation of ADOM with BPMN as ADOM-BPMN. The utilisation of ADOM for reference models in general has been proposed in Reinhartz-Berger et al. (2005) and Soffer et al. (2007). This paper builds on and extends this suggestion for the organisational context, mainly by providing a detailed and formalised process for validating specific models with respect to their organisational reference model. In this paper we assume that the reference model exists and is valid, and we do not discuss the process of constructing that model.

The remainder of the paper is organised as follows. Section 2 informally introduces and demonstrates ADOM-BPMN and its principles. The examples used in this section are of purchase requisition reference model within a university and its instantiation in two sub units: a library and an acquisition department. Section 3 formalises ADOM-BPMN, emphasising the validation of different business process models against the organisational reference model. Section 4 reviews related work and establishes the contribution of the paper. Finally, Section 5 concludes and refers to future research plans.

2 ADOM-BPMN in a nutshell

Reference models capture generic knowledge and hence they introduce additional challenges that are not supported by a process modelling language alone, such as expressing the allowed variability among business processes within a domain or an organisation. When creating a specific model, this variability can be manifested in:

- a omitting model elements which are not relevant for a particular business unit
- b including one or more locally-adapted variants of some model elements
- c introducing specific elements which are not part of the organisational standards, as specified in the reference model.

All these are possible in general, but some restrictions should be made so, e.g., specific elements cannot be omitted, some (partial) order of execution must be maintained, etc. The challenge is to specify such restrictions in the reference model and to be able to verify that they are not violated in the specific models.

To cope with these challenges, we enhanced the BPMN meta-model with two types of classifiers that are added to all BPMN model elements, such as activities, events, gateways and sequence flows. The two types of classifiers are multiplicity indicators and reference model classifiers. Multiplicity indicators are attached to reference model elements and denote the possible lowest and upper-most boundaries of times variants of these elements may appear in a business process model. These are denoted by $\langle \text{min}, \text{max} \rangle$ near the reference model elements. The default multiplicity $\langle 0, n \rangle$ implies no constraints, thus will not explicitly appear in the reference model. The reference model classifiers are associated with elements in the specific (business process) models and denoted by $\langle \text{reference model element name} \rangle$ near the specific model element names. These associations imply that the business process elements are variants of the respective reference model elements.

When a business process is derived from a reference model, the reference model elements are instantiated by the specific (business process) model elements, specialising these elements and providing more information about the specific situation. The association between a reference

model element and its instantiations is mainly done through the reference model classifiers. A single reference model element may have more than one instantiation in a business process model. For example, a quality check element in a reference model may be instantiated by various kinds of quality checks, performed sequentially or concurrently in a specific business process model. The rest of this section elaborates how to represent reference models (Section 2.1) and how to instantiate them (Section 2.2) in ADOM-BPMN. It demonstrates these activities through a purchase requisition process which manages requests for purchasing goods and services in a university. The university employs different purchase requisition processes for various purposes: for buying books, e-journals, software, hardware, office equipment and so on. These processes may differ from each other but they also have to follow the organisational policies, e.g., they are initiated by an explicit request which may be manually or automatically entered. They need different types of approvals, some of which may be timed. They may require quotation handling from suppliers and when successfully finished they result in purchase orders.

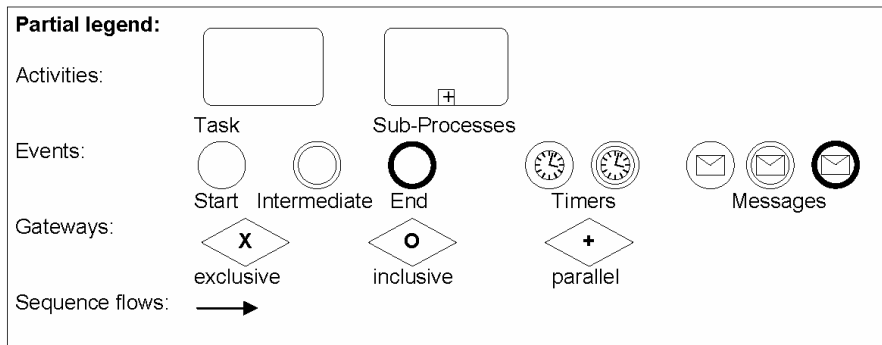
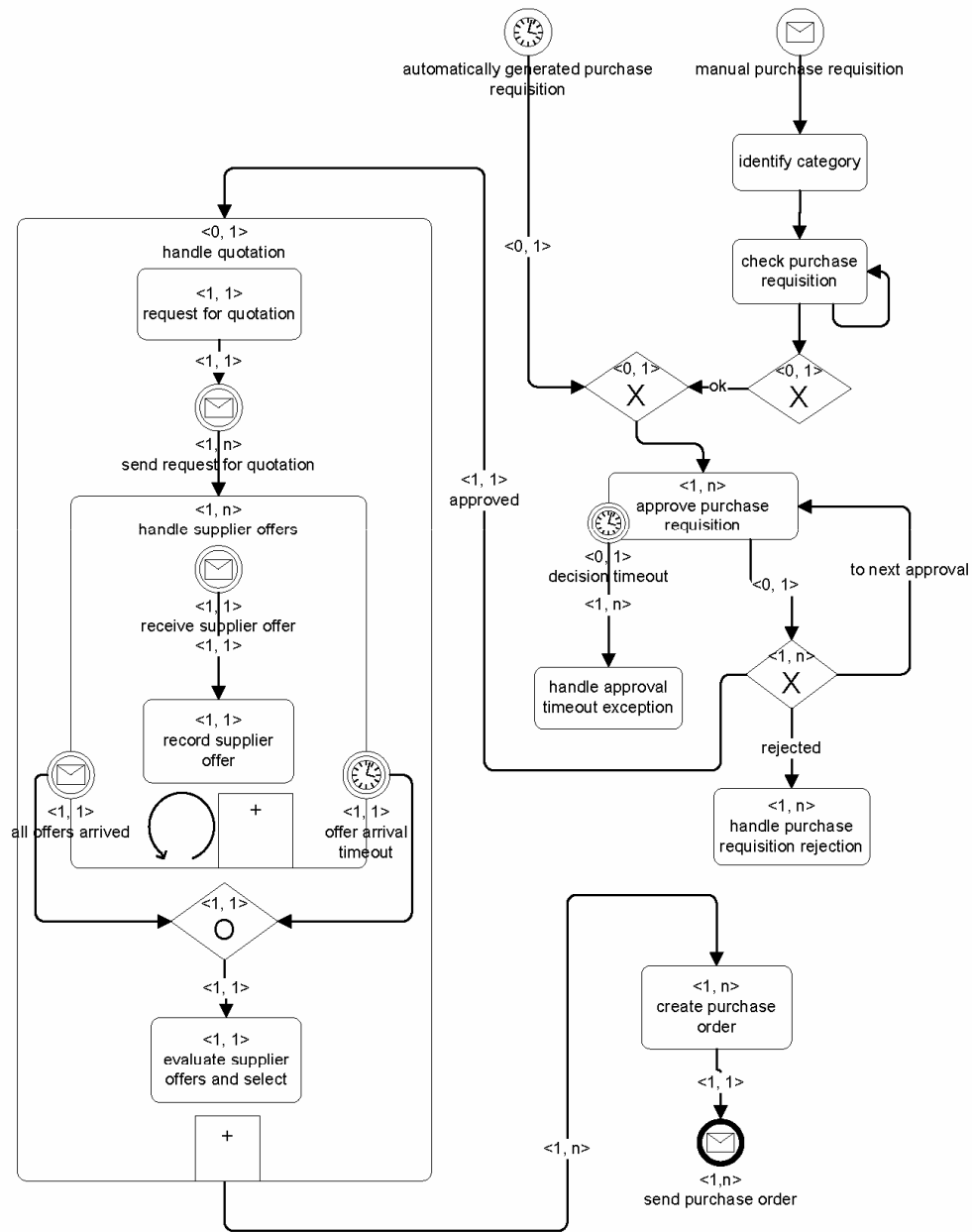
2.1 Representing reference models in ADOM-BPMN

Reference models in ADOM-BPMN are represented as regular models in the modelling language (BPMN in our case). Their aim is to specify the generic knowledge of a domain in terms of common elements (commonality) and allowed variants (variability). As mentioned above, each reference model is (explicitly or implicitly) associated with a multiplicity indicator, thus commonality is expressed as mandatory elements, i.e., elements whose minimal value of multiplicity indicator is 1. Variability is specified in different ways:

- 1 optional elements, i.e., elements whose minimal value of multiplicity indicator is 0
- 2 variants, i.e., elements whose maximal value of multiplicity indicator is greater than 1
- 3 application specific elements, i.e., elements that appear in the specific (business process) model and have no counterparts in the reference model.

Figure 1 depicts the aforementioned purchase requisition reference model in ADOM-BPMN. The process can be either manually or periodically initiated. A (periodical) purchase requisition message can be generated by other systems or business processes in the organisation (university) and triggers the purchase process. The manual initiation starts with an explicit request which may need category identification and a set of checks (e.g., for correctness and necessity). Note that the reference model does not guide the designer how to handle cases in which the purchase requisition checks fail, since these cases are not part of the organisation policies and may be particular to the different departments and their business processes.

Figure 1 The reference (domain) model of the purchase requisition process



After the initiation of the purchase requisition, a set of (at least one) approval activities has to be carried out. The organisation requires that the approval activities will be performed only sequentially for avoiding cases in which different authorities decide oppositely on the same request. Furthermore, an approval activity may be associated with a timeout constraint, meaning that if the approval has not finished within the requested time period, another activity, called ‘handle approval timeout exception’ in the reference model, should be performed.

If a purchase requisition is approved, it may require quotation handling, including:

- 1 Sending the request for quotation to the relevant suppliers.
- 2 Handling (receiving and recording) supplier offers (each purchase requisition process in the university may have more than one way to handle the supplier offers, e.g., according to the supplier, the product or service, and so on).
- 3 Evaluating the supplier offers and selecting the most suitable for the request at hand.

The reference model further specifies that ‘handle supplier offers’ may be an iterative activity and moving to the next stage, ‘evaluate supplier offers and select’, occurs if all offers have arrived or the specified offer arrival time has expired.

After quotation handling or approving the purchase requisition (in case that quotation is not needed), a purchase order is created. A particular business process in the organisation may have several different types of purchase order creation activities. However, each one of them has its own send purchase order (as the <1, 1> sequence flow between the two elements specifies).

2.2 Instantiating reference models in ADOM-BPMN

An *instantiation of a reference model* is a business process model that follows the guidelines of the reference model and fulfils its constraints. Any particular concept in the business process model can be associated with a reference model element via the reference model classifier, implying that the specific element plays the role of the reference model in the business process model. If the name of the reference model element is satisfactory in the context of the business process model, then the element has only reference model classifier (without a name), implying the adoption of the reference element name by the specific process. In the university case, various departments may have different instantiations of the purchase requisition reference model, for example, the library and the acquisition department. In the library, different products can be purchased: journals, books, proceedings and so on. Some of them are electronic and the others are hardcopies. Some of them periodically appear (e.g., journals) and hence, should be automatically

purchased while others are single publications. Some of them are purchased for teaching purposes and others, for research purposes. Figure 2 specifies the purchase requisition process in the university library. This process includes several purchase requisition checks (the availability of the product in the library, as well as the researcher or the library budget), one approval activity (of the library manager) and three types of creating and sending purchase orders according to existing library contracts. The process does not require any quotation handling activities and does not limit the library manager approval in time but it adds some unit-specific tasks (e.g., ‘identify the necessity of the purchase of an available product’). For clarity purposes, the elements that are variants of reference model elements are coloured in grey in the library purchase requisition model while the library-specific elements are white coloured.

A different example of a purchase requisition process in the university is the one used by its acquisition department for purchasing equipment (computers, office materials, and so on). This process, specified in Figure 3, although different than the library purchase requisition process, has to maintain the university policies as well. The purchase requisition checks (for availability and budget) can be performed in parallel. There are three kinds of required approvals. These approvals are timed (between 1 and 2 weeks) and when the timeout is expired, announcements are sent to the relevant authorities, waiting again for their approval. There are quotation handling tasks in which the evaluation and selection activity is refined into ‘quality evaluation’, ‘budget evaluation’ and ‘select supplier’ and finally, a purchase order is created and sent to the supplier. Here again, the organisational reference model-derived elements are coloured in grey.

In order to check that the different business processes are actually valid instantiations of the purchase requisition reference model, ADOM-BPMN should be formalised and a validation procedure should be defined.

3 Formalising ADOM-BPMN and its validation procedure

In this section we formally present ADOM-BPMN, elaborating on its validation procedure.

3.1 ADOM-based instantiation definitions

ADOM’s models include four categories of elements: first-order, dependent, relational and logical, as defined below and further related to BPMN.

Definition 1: (relational, non-relational element): A *relational element* re is a binary directional relationship between two other elements. Notation: $re = (s, t)$ connects a source element s to a destination element t . A *non-relational element* is an element that cannot be expressed as a relational one.

Figure 2 The library purchase requisition process

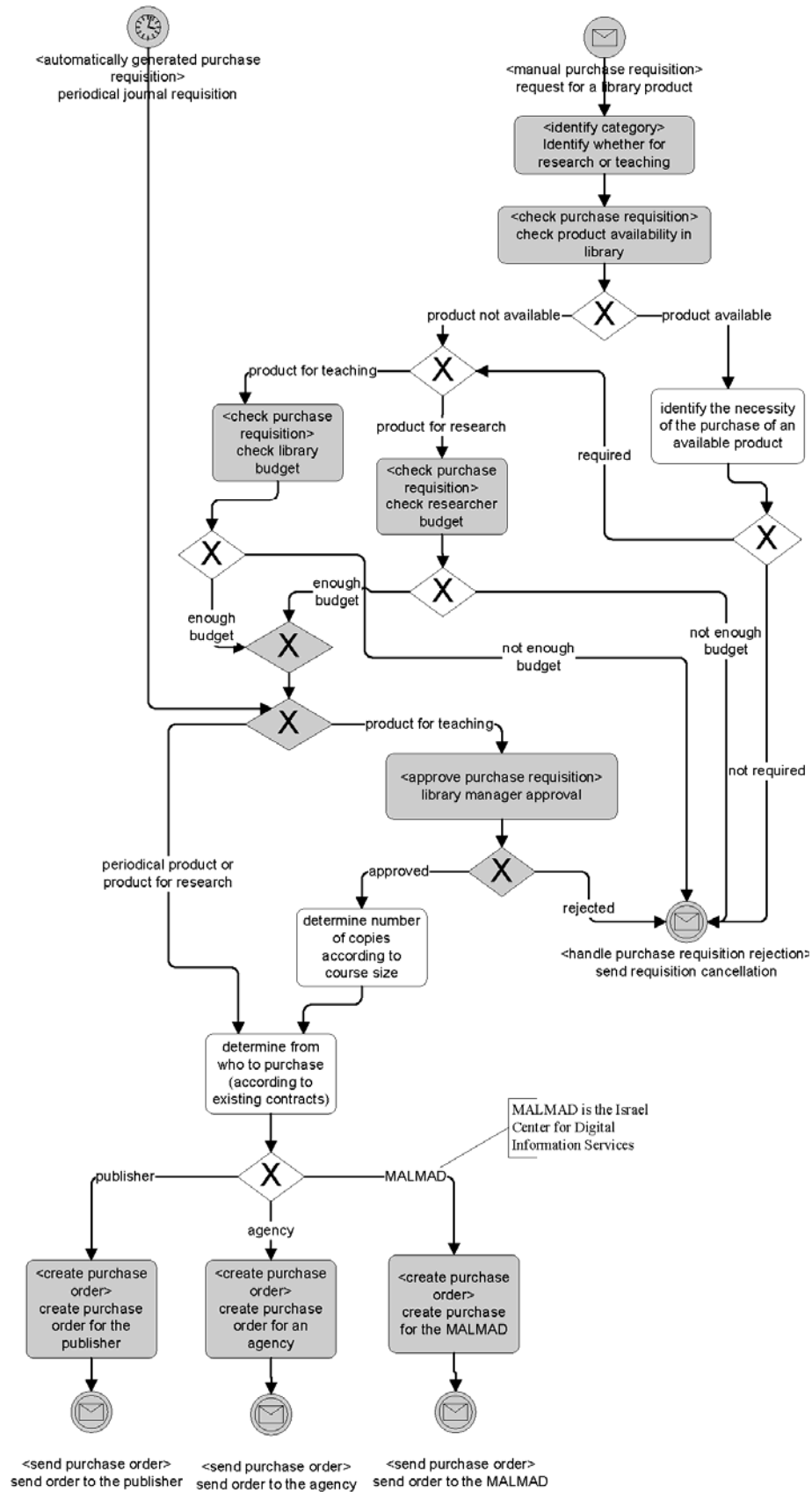
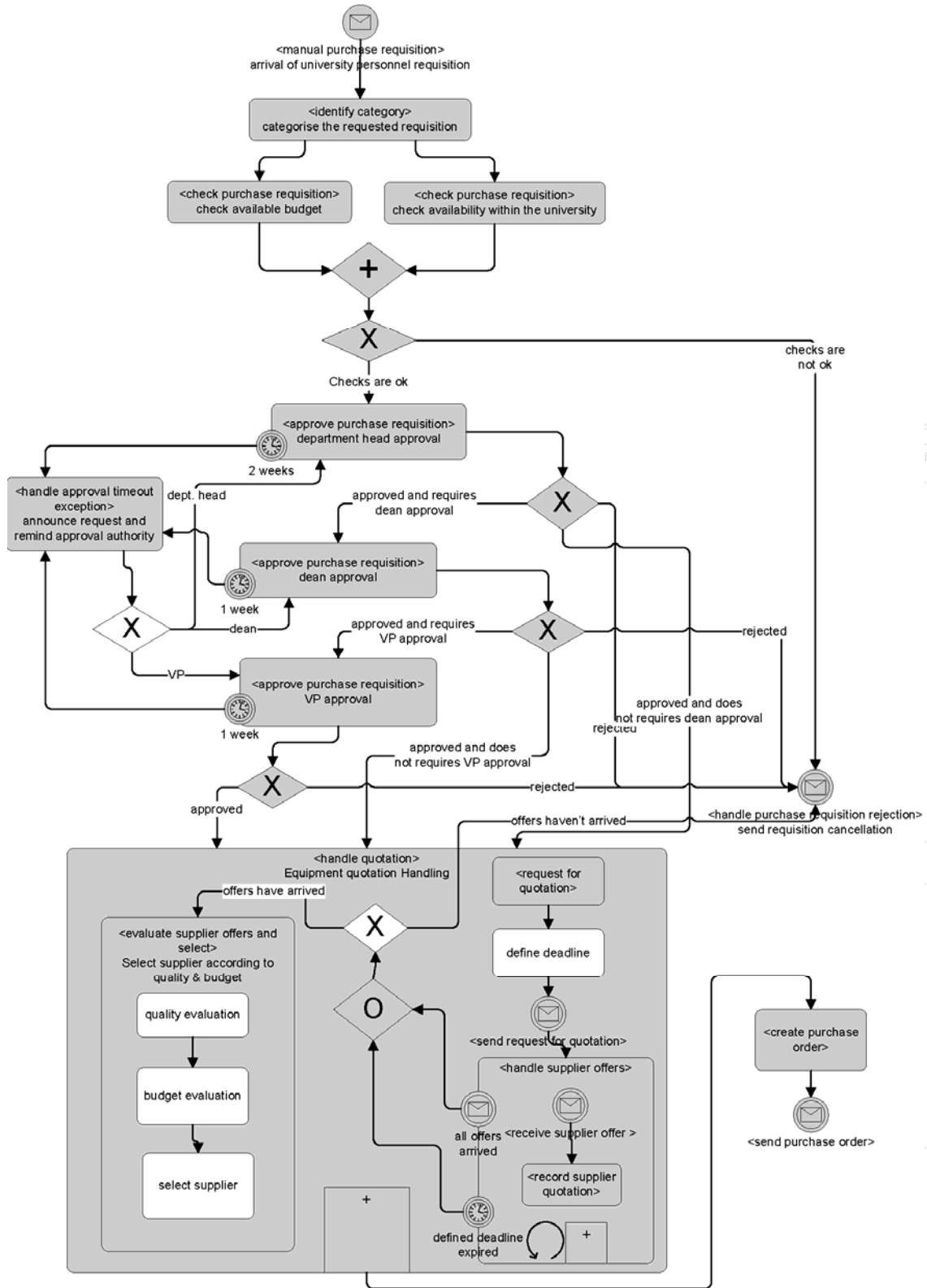


Figure 3 The equipment purchase requisition process



BPMN sequence flows are examples of relational elements, whereas events and tasks are examples of non-relational elements.

Definition 2: (dependent element): A dependent element e in model M is an element whose existence in the model depends on at least one element d in model M .

Notation: $e \dashv_M d$, e depends on the *dependee* d in model M .

In BPMN, tasks and sub-processes that compose other activities are examples of dependent elements, as they depend on their owning sub-processes.

Definition 3: (first order element): A first order element in model M is a non-relational element which is not dependent in model M .

In the case of BPMN, tasks and events on the upper-most level are first order elements.

Definition 4: (reference model): A reference model RM in ADOM-BPMN is a triple $(E_{RM}, MULT, mi)$ such that E_{RM} is a set of model elements in BPMN, $MULT \subseteq \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a set of multiplicity pairs (where \mathbb{N} is the set of the natural numbers and ∞ represents ∞) and $mi: E_{RM} \rightarrow MULT$ is a function. The elements in E_{RM} are termed reference model elements and the elements in $MULT$ are termed multiplicity indicators.

Figure 1 is an example of an ADOM-BPMN reference model that represents a purchase requisition process in a university.

Definition 5 (business process model): A business process model BP in ADOM-BPMN is a triple (E_{BP}, C, cl) such that E_{BP} is a set of model elements in BPMN, C is a set of model elements such that there exists a reference model $RM = (E_{RM}, MULT, mi)$, $C \subseteq E_{RM} \cup \{\text{null}\}$ and $cl: E_{BP} \rightarrow C$ is a mapping. The elements in E_{BP} are termed business process elements and the elements in $C - \{\text{null}\}$ are termed reference model classifiers.

Figure 2 and Figure 3 exemplify two business process models of a library purchase requisition process and an equipment purchase requisition process, respectively.

Definition 6 (element instantiation): Let $BP = (E_{BP}, C, cl)$ be a business process model and $RM = (E_{RM}, MULT, mi)$ – a reference model. A first order or dependent business process element $e_{BP} \in E_{BP}$ *instantiates* a reference model element $e_{RM} \in E_{RM}$ iff:

- 1 The type (meta-class) of e_{BP} is identical to that of e_{RM} .
- 2 $cl(e_{BP}) = e_{RM}$
- 3 If e_{BP} is a dependent element then $\exists d_{BP} \in E_{BP}$, $d_{RM} \in E_{RM}$ such that $e_{BP} \dashv_{BP} d_{BP} \wedge e_{RM} \dashv_{RM} d_{RM} \wedge d_{BP}$ instantiates d_{RM} .

In other words, Definition 6 specifies that the instantiation of first order reference model elements is exclusively defined by the reference model classifiers, while the

instantiation of dependent elements require in addition the element context, i.e., its dependee. As an example, consider the task ‘categorise the requested requisition’ in Figure 3, which is a first order element. Its reference model classifier, ‘identify category’, implies it is an instantiation of the reference model task ‘identify category’. The activity ‘select supplier according to quality and budget’, which is a dependent element in the same figure (it depends on ‘equipment quotation handling’) is an instantiation of ‘evaluate supplier offers and select’, because:

- 1 They are both represented by the same BPMN construct.
- 2 The reference model classifier of ‘select supplier according to quality and budget’ is ‘evaluate supplier offers and select’.
- 3 The dependee of ‘select supplier according to quality and budget’, ‘equipment quotation handling’, is an instantiation of the dependee of ‘evaluate supplier offers and select’, ‘handle quotation’.

The instantiation of relational elements differ as a relational element may be instantiated by a path of elements. The following definitions make this distinction.

Definition 7 (path): A path between two non-relational elements e_s and e_t is a set of distinctive non-relational (first order or dependent) elements $nr_1 \dots nr_m$ and distinctive relational elements re_i ($i = 1 \dots m-1$), such that $\forall i = 1 \dots m-1$ $re_i = (nr_i, nr_{i+1})$. Furthermore, $e_s = nr_1$ is termed the path *source* and $e_t = nr_m$ is termed its *destination*.

As an example of a path consider the following sequence of activities and events in Figure 3: ‘request for quotation’, ‘define deadline’ and ‘send request for quotation’.

Definition 8 (path length): The length of a path is the number of its relational elements.

Note that a relational element together with its source and destination make a path of length 1. The length of the path mentioned above is 2. Unlike first order and dependent elements, which can be respectively instantiated by first order and dependent elements, relational elements can be instantiated by either relational elements or paths.

Definition 9 (path instantiation): A path p_{BP} from s_{BP} to t_{BP} in a business process model $BP = (E_{BP}, C, cl)$ *instantiates* a path p_{RM} from s_{RM} to t_{RM} in a reference model $RM = (E_{RM}, MULT, mi)$ iff:

- 1 \exists non-relational elements $e_{RM}, e'_{RM} \in p_{RM}$, $e_{BP}, e'_{BP} \in p_{BP}$ such that e_{BP} instantiates $e_{RM} \wedge e'_{BP}$ instantiates e'_{RM} . This condition aims at avoiding referring to enterprise-specific paths or non-relational elements as instantiations of optional paths (i.e., paths that all their elements are optional).

- 2 \forall relational element $e_{RM} = (e_{RM}, e'_{RM}) \in p_{RM}$ such that
 $\exists e_{BP}, e'_{BP} \in p_{BP}$ e_{BP} instantiates $e_{RM} \wedge e'_{BP}$ instantiates
 $e'_{RM} \Rightarrow \exists$ a path from e_{BP} to e'_{BP} in p_{BP} .

So far, we ignored BPMN gateways, which are the logical elements that specify the control flow of a process. In order to define model instantiation, we need to introduce these elements to our definitions.

Definition 10 (split logical element, join logical element, logical element, role): A *split logical element* is a mutual source of two or more paths. A *join logical element* is a mutual destination of two or more paths. A *logical element*, is a generalisation of split logical elements and join logical elements. The *role* of a logical element is respectively, split or join.

Four types of gateways are defined in BPMN: exclusive (XOR), inclusive (OR), parallel (AND) and complex. Since the semantics of complex gateways is determined through associating expressions in other (logical) languages (OMG-BMI, 2006), we refer in this paper only to the three basic logical element types used in process modelling languages, namely parallel, inclusive and exclusive.

Definition 11 (logical element type): A *type* of a logical element may be AND, OR or XOR, implying the logical relations between its paths.

Two possible interpretations to reference model logical elements exist. First, the logical element can be used for making decisions when the specific business process model is designed (build-time decision), thus changing the type of the gateway or omitting it from the model is allowed. Alternatively, the gateway can remain in the specific model, leaving the decision to be made at run time. These possibilities are specified as instantiation equivalence of logical elements. The decision on which path to take, the manual or the periodical one, in Figure 1, for example, may be done at design-time or run-time, as the multiplicity indicator $\langle 0, 1 \rangle$ of the exclusive gateway that joins these paths specifies: if this gateway is not included in the business process, then the decision on which path to take is done at design-time (which path to include in the specific business process). Alternatively, if this gateway appears in the business process model, then the decision is left to run-time.

Table 1 Instantiation equivalence matrix

| | | Reference model | | |
|------------------------|-----|-----------------|----|-----|
| | | AND | OR | XOR |
| Business process model | AND | √ | √ | |
| | OR | | √ | |
| | XOR | | √ | √ |

Definition 12 (instantiation equivalent): A type t of a logical element is *instantiation equivalent* to a type t' if a logical element of type t can be considered as an

instantiation of a logical element of type t' . The cell $IE_{t'}$ in Table 1 defines whether a type t is instantiation equivalent to a type t' . Note that instantiation equivalence is necessary but not sufficient for determining if a logical element is instantiated by another logical element, as defined next.

The instantiation of logical elements can only be defined on the basis of path instantiation.

Definition 13 (logical element instantiation): A logical element le_{BP} in a business process model $BP = (E_{BP}, C, cl)$ instantiates a logical element le_{RM} in a reference model $RM = (E_{RM}, MULT, mi)$ iff:

- 1 The roles of le_{BP} and le_{RM} are the same.
- 2 The type of le_{BP} is instantiation equivalent to the type of le_{RM} .
- 3 If le_{RM} and le_{BP} are split logical elements, then $\exists p_{BP1}, p_{BP2} \subseteq E_{BP}, p_{RM1}, p_{RM2} \subseteq E_{RM}$ such that $p_{BP1} \neq p_{BP2} \wedge p_{RM1} \neq p_{RM2} \wedge le_{BP}$ is the source of both p_{BP1} and $p_{BP2} \wedge le_{RM}$ is the source of both p_{RM1} and $p_{RM2} \wedge p_{BP1}$ and p_{BP2} instantiate p_{RM1} and p_{RM2} , respectively.
- 4 If le_{RM} and le_{BP} are join logical elements, then $\exists p_{BP1}, p_{BP2} \subseteq E_{BP}, p_{RM1}, p_{RM2} \subseteq E_{RM}$ such that $p_{BP1} \neq p_{BP2} \wedge p_{RM1} \neq p_{RM2} \wedge le_{BP}$ is the destination of both p_{BP1} and $p_{BP2} \wedge le_{RM}$ is the destination of both p_{RM1} and $p_{RM2} \wedge p_{BP1}$ and p_{BP2} instantiate p_{RM1} and p_{RM2} , respectively.

In other words, a logical element is an instantiation of another logical element of the same type if according to its type, it splits or joins at least two paths which are instantiations of at least two (different) paths that are split or joined by the respective logical element in the reference model. For example, the optional inclusive gateway that appears at the beginning of the reference model in Figure 1 (after the ‘check purchase requisition’ activity) is not instantiated at all in Figure 2 and is instantiated by the parallel gateway connecting ‘check available budget’ and ‘check availability with the university’ to the approval steps in Figure 3.

Note that two or more paths that have a mutual source or destination (namely, a shared logical element) may include other logical elements which connect them to other paths. To make a clear distinction of whether a logical element in a specific model instantiates a reference model logical element, we require that logical elements included in the paths under considerations should not be instantiations of the same reference model logical element.

Finally, a business process model instantiates a reference model if each mandatory reference model element has the required number of business process counterparts, as indicated by the multiplicity indicator constraints.

Definition 14 (model instantiation): A business process model $BP = (E_{BP}, C, cl)$ instantiates a reference model $RM = (E_{RM}, MULT, mi)$ iff:

- 1 \forall non-relational element $e_{RM} \in E_{BP}$ such that e_{RM} is not a logical element $\wedge mi(e_{RM}) = (k, m)$, $k \in \mathbb{N}$, $m \in \mathbb{N} \cup \{n\} \Rightarrow \exists r$ elements $e_{BPi} \in E_{BP}$ ($i = 1, \dots, r$ where $k \leq r \leq m$) such that e_{BPi} instantiates e_{RM} . Note the possibility that $k = 0$, in which case it is possible that $r = 0$ and there are no e_{BPi} .
- 2 \forall relational element $re = (e_{RM1}, e_{RM2}) \in E_{RM}$, $mi(re) = (k, m)$, $k \in \mathbb{N}$, $m \in \mathbb{N} \cup \{n\} \forall e_{BP} \in E_{BP} e_{BP}$ instantiates $e_{RM1} \wedge \exists e'_{BP} \in E_{BP} e'_{BP}$ instantiates e_{RM2} , $\exists r$ paths p_{BPi} in E_{BP} ($i = 1, \dots, r$ where $k \leq r \leq m$) from e_{BP} to e'_{BPi} such that p_{BPi} instantiates $re \wedge e'_{BPi}$ instantiates e_{RM2} .
- 3 \forall relational element $re = (e_{RM1}, e_{RM2}) \in E_{RM}$, $mi(re) = (k, m)$, $k \in \mathbb{N}$, $m \in \mathbb{N} \cup \{n\} \forall e_{BP} \in E_{BP} e_{BP}$ instantiates $e_{RM2} \wedge \exists e'_{BP} \in E_{BP} e'_{BP}$ instantiates e_{RM1} , $\exists r$ paths p_{BPi} in E_{BP} ($i = 1, \dots, r$ where $k \leq r \leq m$) from e'_{BPi} to e_{BP} such that p_{BPi} instantiates $re \wedge e'_{BPi}$ instantiates e_{RM1} .
- 4 \forall logical element $le_{RM} \in E_{RM}$ such that $mi(le_{RM}) = (k, m)$ $k \in \mathbb{N}$, $m \in \mathbb{N} \cup \{n\}$, $\exists r$ logical elements $le_{BPi} \in E_{BP}$ ($i = 1, \dots, r$ where $k \leq r \leq m$) le_{BPi} instantiates le_{RM} .

Figure 2 and Figure 3 are model instantiations of the reference model shown in Figure 1. The grey coloured shapes represent element instantiation. The reference model path ‘identify category’ \rightarrow ‘check purchase requisition’ \rightarrow ‘approve purchase requisition’ \rightarrow ‘handle quotation’ \rightarrow ‘create purchase order’, for example, is instantiated by different specific paths, including:

- 1 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check library budget’ \rightarrow ‘library manager approval’ \rightarrow ‘determine number of copies ...’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase order for the publisher’ in Figure 2
- 2 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check library budget’ \rightarrow ‘library manager approval’ \rightarrow ‘determine number of copies ...’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase order for an agency’ in Figure 2
- 3 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check library budget’ \rightarrow ‘library manager approval’ \rightarrow ‘determine number of copies ...’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase for the MALMAD’ in Figure 2
- 4 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check researcher budget’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase order for the publisher’ in Figure 2

- 5 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check researcher budget’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase order for an agency’ in Figure 2
- 6 ‘identify whether for research or teaching’ \rightarrow ‘check product availability in library’ \rightarrow ‘check researcher budget’ \rightarrow ‘determine from who to purchase ...’ \rightarrow ‘create purchase for the MALMAD’ in Figure 2
- 7 ‘categorise the requested requisition’ \rightarrow ‘check availability within the university’ \rightarrow ‘department head approval’ \rightarrow ‘dean approval’ \rightarrow ‘VP approval’ \rightarrow ‘equipment quotation handling’ \rightarrow ‘create purchase order’ in Figure 3
- 8 ‘categorise the requested requisition’ \rightarrow ‘check available budget’ \rightarrow ‘department head approval’ \rightarrow ‘dean approval’ \rightarrow ‘VP approval’ \rightarrow ‘equipment quotation handling’ \rightarrow ‘create purchase order’ in Figure 3.

As the specific logical expressions may become complex and enterprise-specific activities can be inserted between activities, a validation procedure is required in order to determine whether a specific business process model satisfies the reference model constraints. This procedure is defined in the next sub-section.

3.2 An ADOM-based validation procedure

The validation of a business process consists of three main parts:

- 1 Checking the syntactic and logical correctness of the business process model, e.g., using the modelling language meta-model and supporting tools.
- 2 Validating the semantic correctness of the business process model with respect to the specific unit requirements, e.g., by carrying out design reviews.
- 3 Validating the business process model against the organisational policies.

ADOM supports the last part of the validation, ensuring that business process models adhere with the organisational policies or with the technology infrastructure as reflected in a reference model. When the validation procedure is applied to already existing process models, which were not created as instantiations of the reference model, a preliminary step is required. In the preliminary step, the business process elements are classified according to the reference model elements, and annotated by reference model classifiers.

The ADOM validation procedure includes three steps: element reduction, element unification and model matching.

Element reduction: Since the design of a specific model may entail adding model elements which are specific to the enterprise or to the organisational unit and are not

included in the reference model, the purpose of this step is to remove these elements from the model.

Definition 15 (enterprise-specific elements): The group of *enterprise-specific elements* in a business process model $BP = (E_{BP}, C, cl)$ with respect to a reference model $RM = (E_{RM}, MULT, mi)$ is defined as: $ES(BP, RM) = \{e_{BP} \in E_{BP} \mid (e_{BP} \text{ is a non-relational element} \wedge \neg \exists e_{RM} \in E_{RM} \text{ such that } e_{BP} \text{ instantiates } e_{RM}) \vee (e_{BP} \text{ is a relational element} \wedge \neg \exists \text{paths } p_{BP} \subseteq E_{BP}, p_{RM} \subseteq E_{RM} \text{ such that } e_{BP} \in p_{BP} \wedge p_{BP} \text{ instantiates } p_{RM})\}$.

In the business process models of Figure 2 and Figure 3, the white coloured elements are enterprise-specific.

When an enterprise-specific (or unit-specific) element is removed, some compensating operations should be made, to keep the model consecutive. In particular, these operations relate to relational elements whose sources or destinations are removed.

Definition 16 (compensation): A *compensation for the omission of a non-relational element e in model M* , $co_{e,M}$, is a function from M to a model M' , such that:

- 1 $e' \in M \wedge e' \notin M' \Leftrightarrow e' = e \vee (e' \text{ is a relational element} \wedge (\exists t' \in M \ e' = (e, t') \vee \exists s' \in M \ e' = (s', e))) \vee e' \perp_M e$
- 2 $e' \in M' \wedge e' \notin M \Leftrightarrow e' \text{ is a relational element} \wedge \exists s', t' \in M \text{ such that } e' = (s', t') \wedge (s', e) \in M \wedge (e, t') \in M$

The compensation operations are introduced to the specific business process through a sequence of reduction steps, each of which includes the removal of an enterprise (unit)-specific element. To avoid redundant work, first order elements are removed first and only afterwards dependent and relational elements are removed. Once all the possible reduction steps have been performed, the resulting model is termed the *reduced model*.

Definition 17 (reduction step): A *reduction step i* of a business process model BP with respect to a reference model RM , $R_i(BP, RM)$, is $co_{e_{R_i-1}(BP, RM)}$ where $e \in ES(R_{i-1}(BP, RM), RM) \wedge (e \text{ is relational or dependent} \Leftrightarrow \neg \exists e' \in ES(R_{i-1}(BP, RM), RM) \text{ such that } e' \text{ is first order})$. $R_0(BP, RM) = BP$.

Definition 18 (reduced model): A *reduced model of a business process model BP with respect to a reference model RM* is $R(BP, RM) = R_n(BP, RM)$, where n is the minimal number that satisfies $ES(R_n(BP, RM), RM) = \emptyset$.

In the reduced model of the equipment purchase requisition process (Figure 3), for example, the omission of the activity 'define deadline' implies removing its outgoing and incoming sequence flows and introducing a new sequence flow between the activity 'request for quotation' to the message event 'send request for quotation'.

Element unification: In the element unification step, whose result is termed the *verifiable model*, elements that have the same reference model classifier (in the reduced

model) are unified, leaving only one element for each classifier. The actual multiplicity of these elements is used to denote the number of elements that bear the same classifier in the reduced model. Similarly to the multiplicity indicator in reference models, the actual multiplicity has two values, min and max, which respectively specify the minimal and maximal number of business process elements that are actually classified as the corresponding reference element.

Definition 19 (verifiable model): A *verifiable model of a business process model $BP = (E_{BP}, C, cl)$ with respect to a reference model $RM = (E_{RM}, MULT, mi)$* is a triple $V(BP, RM) = (E_V, actMULT, am)$ such that E_V is a set of model elements, $actMULT \subseteq N \times N$ is a set of multiplicity pairs (where N is the set of natural numbers) and $am: E_V \rightarrow actMULT$ is a mapping.

The verifiable model is constructed from the reduced model as follows:

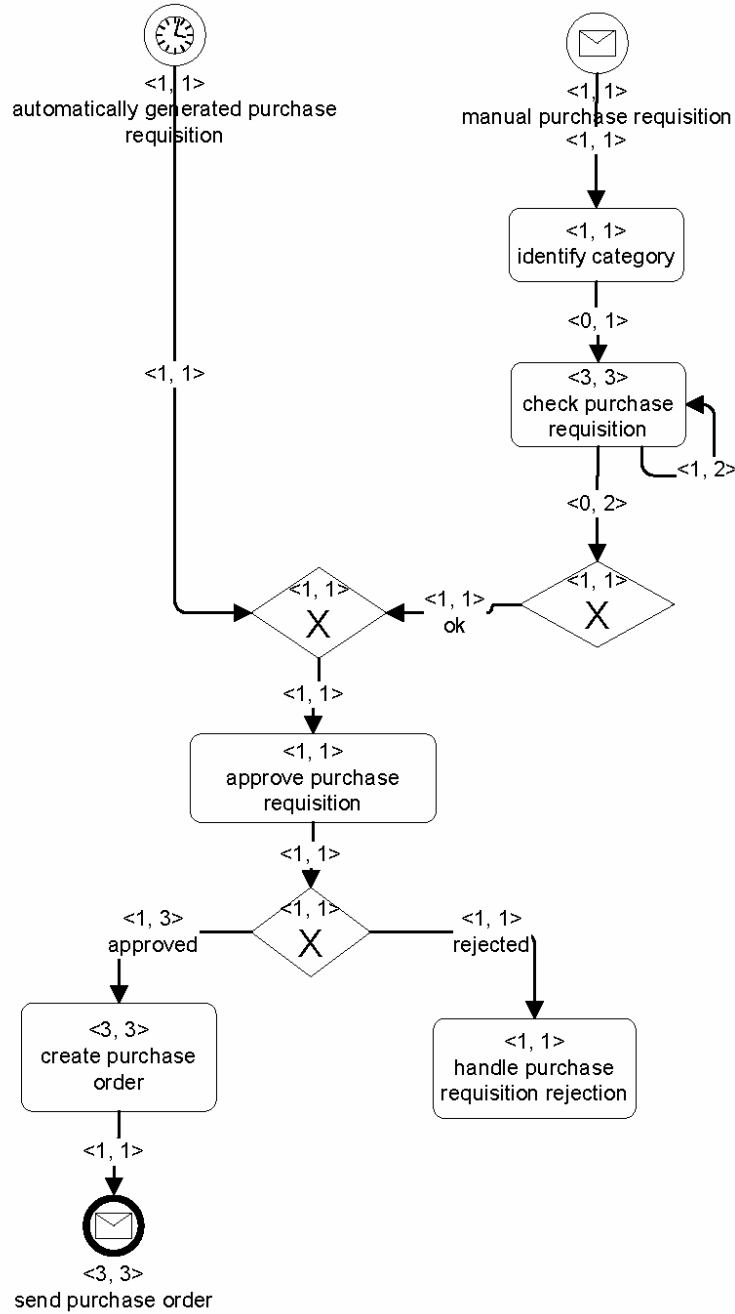
- 1 $\forall e_{R_i} \in R(BP, RM) \ i=1, \dots, k$ such that e_{R_i} is first order and e_{R_i} instantiates the same $e_{RM} \in E_{RM} \ \exists$ a single $e_V \in E_V$ such that $e_V = e_{RM}$ and $am(e_V) = (k, k)$.
- 2 \forall dependent element $e_{R_i} \in R(BP, RM) \ i=1, \dots, k$, $e_{R_i} \perp_{R(BP, RM)} e_i$ such that e_{R_i} instantiates the same $e_{RM} \in E_{RM} \wedge e_i$ instantiate the same $e_{RM}' \in E_{RM}$ (and, hence, are represented by the same $e'_V \in E_V$), \exists a single $e_V \in E_V$ such that $e_V = e_{RM} \wedge am(e_V) = (\min_{e_i} (|\{e_{R_j} \mid e_{R_j} \perp_{BP} e_i\}|), \max_{e_i} (|\{e_{R_j} \mid e_{R_j} \perp_{BP} e_i\}|)) \wedge e_V \perp_{V(BP, RM)} e'_V$
- 3 \forall relational element $e_{R_i} \in R(BP, RM)$, $i=1, \dots, k$, from s_i to t_i such that e_{R_i} instantiates the same $e_{RM} \in E_{RM} \wedge s_i$ instantiate the same $s_{RM} \in E_{RM}$ (and, hence, are represented by the same $s_V \in E_V$) $\wedge t_i$ instantiate the same $t_{RM} \in E_{RM}$ (and, hence, are represented by the same $t_V \in E_V$), \exists a single $e_V \in E_V$ such that $e_V = e_{RM} \wedge e_V = (s_V, t_V) \wedge am(e_V) = (\min_e (|\{t_i \mid (e, t_i) \in E_{BP}\}|, |\{s_i \mid (s_i, e) \in E_{BP}\}|), \max_e (|\{t_i \mid (e, t_i) \in E_{BP}\}|, |\{s_i \mid (s_i, e) \in E_{BP}\}|))$

Note that logical elements can be considered as first order, in case they are in the top level or dependent, in case they are embedded in a process.

Figure 4 and Figure 5 respectively depict the verifiable models (after reduction and unification) of the purchase requisition processes in the library and the acquisition department.

Model matching: In the model matching step the verifiable model is compared to the reference model for adherence with respect to the multiplicity indicators. Model matching checks whether the actual multiplicity of the verifiable model elements are within the boundaries of the multiplicity indicators of the corresponding reference model elements. In addition, it verifies that reference model elements that do not appear in the verifiable model are optional.

Figure 4 The verifiable model of the library purchase requisition process



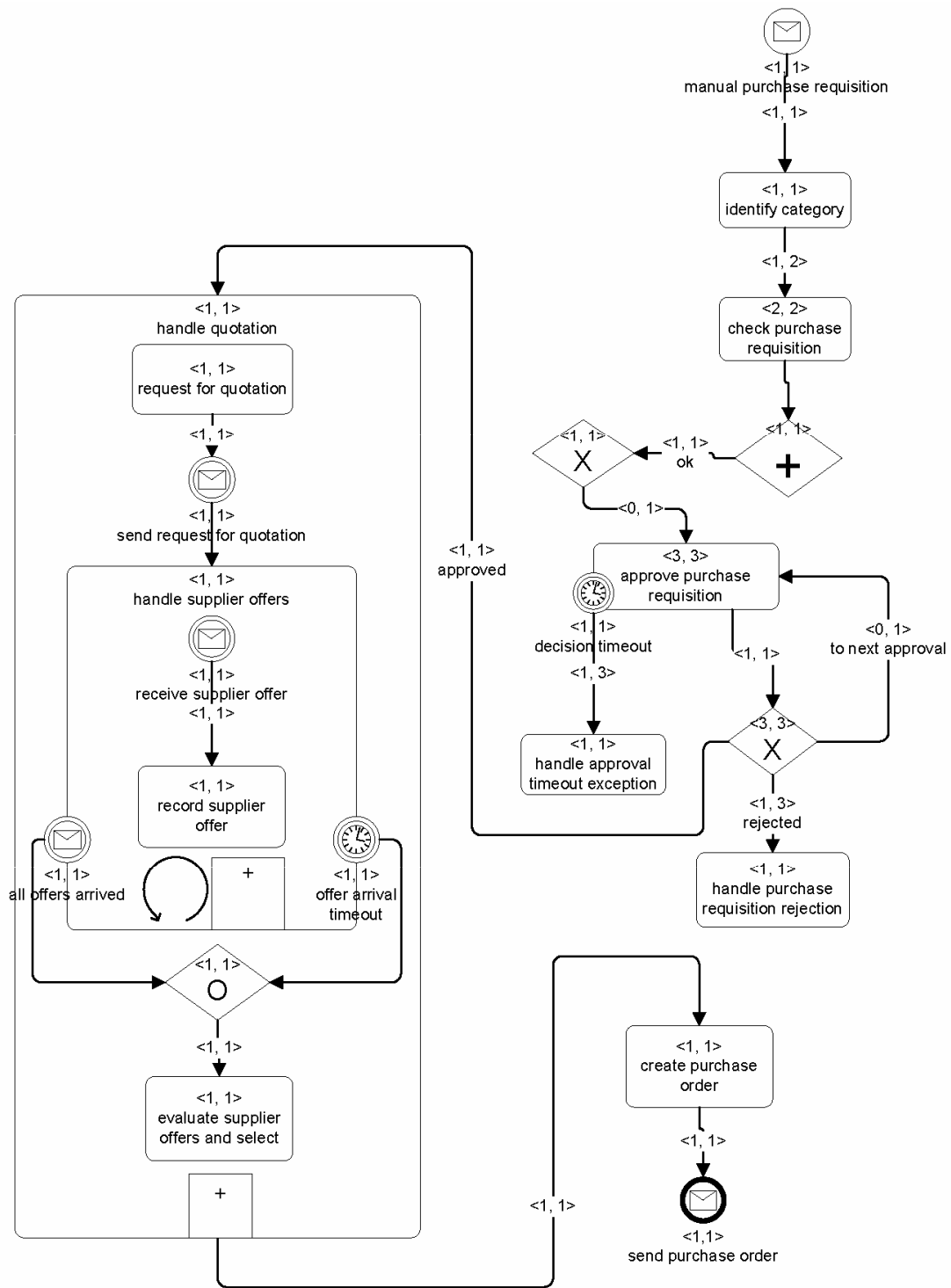
Definition 20 (validation): A business process model $BP = (E_{BP}, C, cl)$ is *valid with respect to* a reference model $RM = (E_{RM}, MULT, mi)$ iff its verifiable model $V(BP, RM) = (E_V, actMULT, am)$ satisfies:

$$1 \quad \forall e \in E_V \exists e' \in E_{RM}, a_{min}, a_{max}, m_{min} \in \mathbb{N}, m_{max} \in \mathbb{N} \cup \{n\} \text{ such that } e = e' \wedge am(e) = (a_{min}, a_{max}) \wedge mi(e') = (m_{min}, m_{max}) \wedge a_{min} \geq m_{min} \wedge a_{max} \leq m_{max}$$

$$2 \quad \forall \text{first order element } e' \in E_{RM} (\neg \exists e \in E_V e = e' \Leftrightarrow \exists m_{max} \in \mathbb{N} \cup \{n\} mi(e') = (0, m_{max}))$$

As can be seen, the verifiable models in Figure 4 and Figure 5 match the purchase requisition reference model depicted in Figure 1 and hence, comply with the organisational policy as determined in the purchase requisition reference model.

Figure 5 The verifiable model of the equipment purchase requisition process



4 Related work

Process management in distributed organisations has mainly been addressed in the context of global ERP implementations (e.g., Clemons and Simon, 2001; Rebstock and Selig, 2000; Yamin and Sinkovics, 2007). The strategy taken for the ERP configuration (and the related business process design) refers to the nature of the globally-distributed organisations. Four such types are identified, global, international, multi-national and transnational (Bartlett and Ghoshal, 1989), varying from each other in the level of central control and required coordination among the units. In Rebstock and Selig (2000) three possible strategies for business process design in a globalised ERP implementation are described: fully centralised, fully distributed, and coordinated. They advocate for the third one, in which processes are first locally analysed. This analysis yields a ‘corporate best practice catalog’ that integrates specific requirements of the local units. The catalog is comparable to an organisational reference model, and its processes are adopted by the local units along with specific processes meeting their own requirements. However, no details are given as to how this catalog is specified and how local adaptations are facilitated.

The need to specifically adapt processes in different organisational units exists also in organisations that are not globally distributed. In Becker et al. (2007) a possible adaptation of a reference model to different organisational units in such settings is discussed. However, no mechanism is proposed to verify that the organisational standards are maintained.

Business process reference models have been discussed, classified and evaluated using a number of evaluation frameworks and criteria (e.g., Fettke and Loos, 2003a, 2003b, 2005; Fettke et al., 2005; Mistic and Zhao, 2000; Schuette and Rothhowe, 1998). Classifying the reuse processes employed by reference models, four types of reuse are distinguished (Reinhartz-Berger et al., 2005): reuse by adoption (e.g., Curran and Ladd, 1999; Scheer, 1998), reuse by assembly (e.g., Lang et al., 1997; Van Es, 1998), reuse by configuration (e.g., Mendling et al., 2005; Recker et al., 2005; Rosemann and Aalst, 2007) and reuse by specialisation (e.g., Stephens, 2001). These types differ from one another in the abstraction level of the reference model, the possible variability of the specific models and the guidance provided by the model to the possible instantiation operations that can be performed. While reuse by configuration is the only type to provide explicit instantiation guidance, reuse by specialisation supports a superior level of variability in the specific processes (Reinhartz-Berger et al., 2005). The reuse type supported by ADOM-BPMN is reuse by specialisation and configuration. However, unlike specialisation-based reference models, ADOM-BPMN explicitly provides reuse guidance. In addition, the validation procedure presented in this paper is novel, since none of the four discussed reuse processes entails validation of the specific processes against the reference model. Note that

Configurable EPC (C-EPC) guides the creation of syntactically lawful EPCs as specific models (Recker et al., 2005) and employs configuration patterns to ensure that the configuration operations semantically follow the configuration specification of the model. In contrast, we provide a semantic validation procedure, in which a business process is validated against the organisational reference model for compliance with its specified constraints.

In the general context of process models, the issue that has been extensively investigated is verification of a model in terms of syntactical and logical errors. Examples of such approaches are soundness verification of Petri-Nets (Aalst, 1997) and EPC (Aalst, 1999), and the application of automated model checkers (Eshuis et al., 2002; Latella et al., 1999). Addressing the business content of process models in general, validation has hardly been proposed in the literature so far. One exception is Speck et al. (2003), which deals with reference modelling and validation of business processes. It was applied to Scheer’s (1998, 1999) models, utilising model checkers. It uses rules and patterns that should be provided in addition to the reference model in order to validate the specific created business processes. Validation of a process model with respect to its ability to attain its goal is proposed in Soffer and Wand (2004). However, the validation in this approach is with respect to its function (goal) only, rather than to its complete behaviour, as we propose in this paper.

5 Conclusions

Reference models are models whose aim is to capture domain knowledge and assist in the design of enterprise specific processes. In this paper we focused on their use for meeting the specific challenges of adequate process design in complex organisations, which entails compliance of the designed processes with some organisational standards. We proposed ADOM as a platform for organisational reference models, and introduced a novel validation procedure to check the compliance of the specific processes with the organisational reference model. BPMN was used as a modelling language for defining activities, events, sequence flows and logical gateways. Note that BPMN has additional types of elements, such as pools, lanes and data objects. However, in this paper, we concentrated on the core elements that are common in many business process modelling languages.

We have already applied a simple version of ADOM-BPMN in an industry-related project implementing service-oriented systems for business processes that originated from a set of reference models. Each activity within the reference model is associated with a set of web service operations, which were carefully selected based on their functional semantics and non-functional properties, as candidates for implementing the specific activity. Moreover, these services are already

synchronised among themselves and their required data mediators are defined. Thus, when checking and enforcing the adherence of the business processes with their corresponding reference model, the development efforts are reduced. The importance of the validation in this situation is that in case of deviation from the reference model, the entire integration and composition of the services should be re-designed as well as the required data mediators.

In future, we intend to resolve issues such as incorporating multiple abstraction levels, as well as to examine the integration of various enterprise model views as part of a reference model.

References

- Aalst, W.M.P. v.d. (1997) 'Verification of workflow nets', *Application and Theory of Petri Nets*, pp.407–426, Springer-Verlag, LNCS 1248, Berlin.
- Aalst, W.M.P. v.d. (1999) 'Formalization and verification of event-driven process chains', *Information and Software Technology*, Vol. 41, No. 10, pp.639–650.
- Bartlett, C.A. and Ghoshal, S. (1989) *Managing Across Borders: The Transnational Solution*, Harvard Business School Press, Boston, MA.
- Becker, J., Delfmann, P. and Knackstedt, R. (2007) 'Adaptive reference modelling: integrating configurative and generic adaptation techniques for information models', in Becker, J. and Delfmann, P. (Eds): *Reference Modelling*, pp.27–58, Springer-Verlag, Berlin.
- Clemons, S. and Simon, S.J. (2001) 'Control and coordination in global ERP configuration', *Business Process Management Journal*, Vol. 7, No. 3, pp.205–215.
- Curran, T.A. and Ladd, A. (1999) *SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain Management*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.
- Czarnecki, K. and Eisenecker, U. (2000) *Generative Programming: Methods, Tools and Applications*, Addison-Wesley Professional.
- Eclipse (2007) available at <http://www.eclipse.org/stp/bpmn/index.php>.
- Eshuis, R., Jansen, D.N. and Weiringa, R. (2002) 'Requirements-level semantics and model checking of object-oriented statecharts', *Requirements Engineering*, Vol. 7, pp.243–263.
- Fettke, P. and Loos, P. (2005) 'Ontological analysis of reference models', in Green, P. and Rosemann, M. (Eds): *Business Systems Analysis with Ontologies*, pp.56–81, Idea Group.
- Fettke, P. and Loos, P. (2003a) 'Classification of reference models – a methodology and its application', *Information Systems and e-Business Management*, Vol. 1, No. 1, pp.35–53.
- Fettke, P. and Loos, P. (2003b) 'Multiperspective evaluation of reference models – towards a framework', in Jeusfeld, M. and Pastor, O. (Eds): *Proceedings of ER'03 Workshops*, pp.80–89, LNCS 2814, Springer-Verlag, Berlin.
- Fettke, P., Loos, P. and Zwicker, J. (2005) 'Business process reference models: survey and classification', *Proceedings of First International Workshop on Business Process Reference Models*, pp.1–15, Nancy, France.
- Gomaa, H. (2005) *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*, Addison-Wesley.
- Intalio (2007) Available at <http://www.intalio.com>.
- Kang, K. C., Kim, S., Lee, J., Kim, K., Shin, E. and Huh, M. (1998) 'FORM: a feature-oriented reuse method with domain-specific reference architectures', *Annals of Software Engineering*, Vol. 5, No. 1, pp.143–168.
- Kang, K., Cohen, S., Hess, J., Novak, W. and Peterson, A. (1990) *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, CMU/SEI-90-TR-021 ADA235785.
- Lang, K., Glunde, J. and Bodendorf, F. (1997) 'A framework for reusable reference process building blocks', *ACM SIGGROUP Bulletin*, Vol. 18, No. 1, pp.68–70.
- Latella, D., Majzik, I. and Massink, M. (1999) 'Automatic verification of a behavioural subset of UML statechart diagrams using the spin model-checker', *Formal Aspects of Computing*, Vol. 11, pp.637–664.
- Mendling, J., Recker, J., Rosemann, M. and Aalst, W. v.d. (2005) 'Towards the interchange of configurable EPCs', in Desel, J. and Ulrich, F. (Eds): *Proceedings of Enterprise Modelling and Information Systems Architectures Workshop*, pp.8–21, Klagenfurt, Austria.
- Misic, V.B. and Zhao, J.L. (2000) 'Evaluating the quality of reference models', *Proceedings of ER 2000 – 19th Conference on Conceptual Modelling*, pp.484–498.
- Moon, M., Yeom, K. and Chae, H.S. (2005) 'An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line', *IEEE Transactions on Software Engineering*, Vol. 31, No. 7, pp.551–569.
- OMG-BMI (2006) *Business Process Modelling Notation Specification*.
- Ouyang, C., Aalst, W.M.P. v.d., Dumas, M. and ter Hofstede, A.H.M. (2006) 'From BPMN process models to BPEL web services', *Proceedings of the 4th International Conference on Web Services (ICWS)*, pp.285–292.
- Rebstock, M. and Selig, J. G. (2000) 'Development and implementation strategies for international ERP software projects', *Proceedings of the 8th European Conference on Information Systems (ECIS2000)*, Vol. 2, pp.932–936.
- Recker, J., Rosemann, M., Aalst, W. v.d. and Mendling, J. (2005) 'On the syntax of reference model configuration – transforming the C-EPC into lawful EPC models', *Proceedings of First International Workshop on Business Process Reference Models*, pp.60–74, Nancy, France.
- Reinhartz-Berger, I. and Sturm, A. (2007) 'Enhancing UML models: a domain analysis approach', *Journal on Database Management (JDM)*, Special issue on UML topics, Vol. 19, No. 1, pp.74–94.
- Reinhartz-Berger, I., Soffer, P. and Sturm, A., (2005) 'A domain engineering approach to specifying and applying reference models', *Proceedings of Enterprise Modelling Information Systems Architecture (EMISA'05)*, Klagenfurt, Austria.
- Rosemann, M. and Aalst, W.M.P. v.d. (2007) 'A configurable reference modelling language', *Information Systems*, Vol. 32, No. 1, pp.1–23.
- Scheer, A. W (1998) *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer Berlin.
- Scheer, A.W. (1999) *ARIS – Business Process Frameworks*, Springer, Berlin.

- Schuette, R. and Rothowe, T. (1998) 'The guidelines of modelling – an approach to enhance the quality in information models', *Proceedings of ER 1998 – 17th International Conference on Conceptual Modelling*, pp.240–254.
- Soffer P. and Wand, Y. (2004) 'Goal-driven analysis of process model validity', *Advanced Information Systems Engineering (CAiSE'04)*, pp.521–535, LNCS 3084, Springer-Verlag, Berlin.
- Soffer, P., Reinhartz-Berger, I. and Sturm, A. (2007) 'Facilitating reuse by specialization of reference models for business process design', *Workshop on Business Process Modelling, Design and Support (BPMDS'07), Proceedings of CAiSE'07 Workshops*, pp.339–348.
- Speck, A., Pulvermüller, E. and Heuzeroth, D. (2003) 'Validation of business process models', *Proceedings of the Workshop on Correctness of Model-Based Software Composition (CMC), In Association with the 17th European Conference on Object-Oriented Programming (ECOOP) 2003*, pp.75–83, Report No. 2003-13, Universitaet Karlsruhe, Fakultae fuer Informatik.
- Stephens, S. (2001) 'Supply chain operations reference model version 5.0: a new tool to improve supply chain efficiency and achieve best practice', *Information Systems Frontiers*, Vol. 3, No. 4, pp.471–476.
- Sturm, A. and Reinhartz-Berger, I. (2004) 'Applying the application-based domain modelling approach to UML structural views', *the 23rd International Conference on Conceptual Modelling (ER'2004)*, pp.766–779, LNCS 3288.
- Thomas, O. (2005) 'Understanding the term reference model in information systems research: history, literature analysis and explanation', *Proceedings of First International Workshop on Business Process Reference Models*, pp.16–28, Nancy, France.
- Van Es, R. (1998) *Dynamic Enterprise Innovation*, Baan Business Innovation B.V., the Netherlands.
- Yamin, M. and Sinkovics, R. R. (2007) 'ICT and MNE reorganisation: the paradox of control', *Critical Perspectives on International Business*, Vol. 4, No. 4, pp.322–336.