

Facilitating Reuse by Specialization of Reference Models for Business Process Design

Pnina Soffer¹, Iris Reinhartz-Berger¹, Arnon Sturm²

¹Department of Management Information Systems,
University of Haifa, Haifa 31905, Israel
spnina@is.haifa.ac.il, iris@mis.haifa.ac.il

²Department of Information Systems Engineering,
Ben-Gurion University of the Negev, Beer Sheva 84105, Israel
sturm@bgu.ac.il

Abstract. Most reference models do not explicitly guide their reuse process. However, they are all designed according to (at least) an implicit intended reuse process. An important type of reuse is specialization, which allows a high level of flexibility and variability when instantiating reference models to specific business processes. In this paper we propose an approach that facilitates reuse by specialization by using a domain engineering approach called ADOM. We identify five possible specialization operations, namely refinement, sub-typing, contextual adoption, omission, and inclusion, discuss their essence, and exemplify their usage within ADOM. These operations and their application within ADOM may provide a significant improvement in business process design while leveraging existing business process reference models.

1 Introduction

The design of business processes can benefit from reusing existing knowledge. The benefits of reuse have been long recognized. These include saving time and resources, reducing development cost, and increasing reliability. Reusable knowledge that can be used for process design exists in the form of reference models, which are models that aim at supporting the construction of enterprise-specific process models [11].

Much attention has been given over the years to the construction of reference models and to the knowledge that is captured in them (e.g., [3]). However, the actual process in which this knowledge is to be reused for business process design has not been widely addressed until recently. In particular, most reference models provide little support (if any) to their actual application in a specific organization.

This paper addresses the reuse process of reference models. We build on knowledge of reuse processes gained within the discipline of domain engineering [1] and apply it to the reuse process of reference models. Specifically, we adopt the

Application-based Domain Modeling (ADOM) approach ([4], [5], [10]), whose reuse process is by specialization (in the wide sense), to reference models. We investigate specialization operations of a process model and how these can be applied to a reference model when designing an enterprise-specific process model.

The rest of the paper is organized as follows. Section 2 reviews and categorizes existing reuse processes of reference models, whereas Section 3 generally introduces the Application-based Domain Modeling approach. In Section 4 we elaborate on the specialization operations in the context of business process design and in Section 5 we demonstrate their use within the ADOM approach. Finally, in Section 6 we conclude and set the basis for future research.

2 Types of reuse processes

As already mentioned, most reference models do not explicitly guide their reuse process. However, they are all designed according to (at least) an implicit intended reuse process, which can be by adoption, by assembly, by configuration, or by specialization. These types vary from each other in the abstraction level of the knowledge captured in the reference model, in the support provided to variability among enterprises applying the reference model, and in the guidance given to the reuse process, as summarized in Table 1.

Table 1: Types of reuse processes in reference models

Reuse type	Example reference	Abstraction level	Variability support and limitations	Reuse guidance
By adoption	[8]	Low	No support	None
By assembly	[12]	Low	Selecting process segments and assembling them; limited to available model segments	None
By configuration	[6]	Low	Including or omitting process parts; limited to available configuration options of the model	Configurable functions and connectors, configuration patterns
By specialization	[7]	High	Specializing, omitting and adding process parts; unlimited	None

As the knowledge captured in a reference model can contribute to the design of enterprise-specific business processes, we argue that the properties that are most influential regarding the adequate use of these models are variability support and reuse guidance. The importance of variability support lies in the fact that processes in different organizations are not identical. A reference model should support variability in order to be applicable in a variety of organizations, so that the resulting specific models meet the particular needs of each organization. Reuse guidance is needed to

support the adaptation process so that the resulting specific model maintains the business logic of the reference model and does not include inconsistencies caused by specific changes and adaptations made.

Considering the reuse types presented in Table 1, reuse by adoption provides no variability support, reuse by assembly and by configuration provide a limited variability support, and reuse by specialization provides an unlimited one. Reuse guidance is provided only by the reuse by configuration approach. No guidance is provided for selecting and integrating model segments in reuse by assembly, and no guidance is given for specializing a generic model in reuse by specialization.

Since, as show in Table 1, reuse by specialization is the only approach that does not pose limitations on the possible variability, this paper addresses reuse by specialization, and aims at providing the reuse guidance which is needed in order to ensure the adequate design of specific processes on the basis of reference models that take this reuse approach.

3 The Application-based Domain Modeling (ADOM) Approach

The Application-based Domain Modeling (ADOM) is rooted in the domain engineering discipline, which is concerned with building reusable assets on one hand and representing and managing knowledge in specific domains on the other hand.

The architecture of ADOM ([5], [10]) consists of three layers: application, domain, and language. The *application layer* consists of models of particular enterprises, including their structure (data) and behavior (business processes). The *language layer* includes metamodels of modeling languages, such as UML, EPC, or others. The intermediate *domain layer* consists of specifications of various application families (domains) and reference models. The ADOM approach enforces constraints among the different layers, and in particular the domain layer (the reference models) enforces constraints on the application layer (the enterprise-specific process models).

ADOM is a generic approach and can be applied to a variety of modeling languages. ADOM's reuse process is mainly done by specialization and configuration. However, as opposed to existing reference models whose intended use is by specialization, in ADOM the constraints enforced by the domain layer provide guidance to the reuse process.

ADOM addresses three main issues: representation of reference models, construction of enterprise-specific process models through specialization operations, and validation of the enterprise-specific models against the relevant reference models. The representation of reference models relates to ways of representing the commonality and variability of various processes that belong to the same category (e.g., business segment) and constraining their specializations. The construction of specific process models relates to the possible operations that can be made on the reference models in order to make them suitable for the enterprises at hand. The resultant process model is termed an *instantiated model* and the whole process is termed an *instantiation* of the reference model. The validation of an instantiated

model verifies that it complies with the constraints specified in the reference model. In this paper we mainly address the second issue of identifying the possible specialization operations of a reference model for the creation of an enterprise-specific process model. Since these operations are derived from the expressiveness of the reference models, we also refer to the first issue of representing reference models in ADOM. The third issue of validation is discussed in [4].

4 Specialization operations for business process design

Specialization can be defined as "a reduction in generality, the act of making something suitable for a special purpose" [9]. In the context of business process design, we can refer to specialization of the basic building blocks activities and their flows. In some process modeling languages the flows are further divided into events, triggers, inputs, outputs, and so on. However, due to space limitation, the focus of this paper is on the specialization operations that can be applied mainly to activities. In Section 4.1 we define specialization operations that can be applied to either atomic or composite activities, while in Section 4.2 we introduce additional operations for whole business processes, which can be viewed as composite activities, composed of atomic and composite activities and their flows.

4.1 Specialization of activities

In order to define a specialization of an activity, we need to start by defining a state, and its change through an activity.

*Definition 1: Consider a part of the world. Its **state** is the set of values of all its properties at a moment in time.*

Since it is not practically possible to address the entire set of properties possessed by the part of the world on which one focuses, for practical purposes we may relate to sets of states.

*Definition 2: A **set of states** is specified by restricting the values of a subset of the properties, while the remaining properties may assume any value.*

For example, "Order is opened" is a set of states, defined by the value of the status property of order, where other properties (e.g., customer, ordered item, etc.) may have any valid value. The definition of an activity relates to sets of states as follows.

*Definition 3: Let A and B be sets of states, such that: (a) $A \cap B = \emptyset$ and (b) given predicates C_1 and C_2 , $A = \{s \mid C_1(s) = \text{TRUE}\}$, $B = \{s \mid C_2(s) = \text{TRUE}\}$. An **activity** $t(A, B)$ is a transition from a state $s_1 \in A$ to a state $s_2 \in B$.*

C_1 and C_2 are respectively termed *pre- and post-conditions* of t , while A and B are respectively termed *initial and final sets* of t . The transition is achieved by some *execution path* f .

*Definition 4: Let $t(A, B)$ and $t'(A', B')$ be activities. $t'(A', B')$ is a **specialization** of $t(A, B)$ if $A' \subseteq A$ and $B' \subseteq B$.*

Note that if $A' \subset A$ and $B' \subset B$ then $t'(A', B')$ is an actual specialization of $t(A, B)$, but if $A' = A$ and $B' = B$ it may be the same activity.

We identify three possible specialization operations for an activity: refinement, sub-typing, and contextual adoption.

Definition 5: Refinement is a change in the granularity of the model, where $t(A, B)$ is replaced by a flow of activities $t_1(A_1, B_1), t_2(A_2, B_2), \dots, t_n(A_n, B_n)$, such that $A_1 \subseteq A$ and $B_n \subseteq B$.

Note that each one of the activities $t_1(A_1, B_1), t_2(A_2, B_2), \dots, t_n(A_n, B_n)$ is not a specialization of $t(A, B)$. As an example, consider a delivery activity, which can be refined to loading, shipping, and unloading. The initial set of this flow of activities includes states where goods need to be delivered and the final set includes states where goods are received by the customer.

Sub-typing deals with defining specific ways of achieving the activity transition.

Definition 6: Let $t(A, B)$ be an activity and f be the execution path associated to it. $t'(A, B)$ is a **sub-type** of $t(A, B)$ if (1) $t'(A, B)$ is a specialization of $t(A, B)$ and (2) its associated execution path, f' , satisfies $f' \neq f$.

Note that different sub-types may have slightly different initial and final sets. However, the main difference between sub-types would be in the ways they are achieved rather than in their pre- and post-conditions. As an example, consider the activity "insert order", which is a transition from a state satisfying "a customer wants to order" to a state satisfying "an order is opened". It can be specialized to the following three sub-types: "insert order via the web", which is a transition from "a customer wants to order via the web" to "a web order has been registered", and "insert order by fax or phone", which is a transition from "a customer wants to order by fax" or "a customer wants to order by phone" to "a manual order has been registered".

Contextual adoption deals with taking the activity and adopting it into a specific context. Note that differently from sub-types which deal with applying similar functionality to the same subjects but in different ways, contextual adoption deals with applying the same (possibly customized) operations to different subjects.

Definition 7: Let $t(A, B)$ be an activity and f be the execution path associated to it. $t'(A', B')$ is a **contextual adoption** of $t(A, B)$ if (1) $t'(A', B')$ is a specialization of $t(A, B)$ and (2) f is the execution path associated to $t'(A', B')$ too.

In other words, different contextual adoptions relate to different initial and final sets, while the way this transformation is achieved should be the same. For example, the activity of checking availability is a transition from a state where the availability of something is unknown to a state where it is known. It can be applied to raw materials as well as to packaging materials and human resources.

Note that we defined the above operations separately. In practice more than one specialization operation can be made to the same activity. For example, a sub-type of an activity can also be refined.

4.2 Specialization of business processes

When a business process, which is composed of atomic and composite activities and flows among them, is specialized, operations are made to both its activities and their flows. In addition to the above discussed operations, further operations are allowed for composite activities in general and business processes in particular. These operations, which are omission and inclusion, increase the variability allowed by a particular reference model.

Omission deals with removing internal activities of the composite activity, which may be irrelevant to the enterprise at hand. Omitting internal activities is possible only if they are optional and their post-conditions are not required by other internal activities within the composite one. *Inclusion*, on the other hand, deals with adding new internal activities which do not appear in the composite reference model activity, but are required for the completeness of the specialized business process.

As demonstrated in the next section, the flow of the specialized activities should follow the flow of their generic counterparts in the reference model, since their pre- and post-conditions, triggers, branching points, and synchronization points should correspond to each other.

5 Specialization of ADOM-based reference models

Having introduced the various specialization operations, in this section we present their usage within ADOM. ADOM includes a mechanism of multiplicity indicators, which are attached to activities and other reference model elements. A multiplicity indicator of an activity specifies a range for the number of specializations of that activity that may be included in an instantiated model. The multiplicity indicator has a lower and an upper constraint (denoted as $\langle \text{min}, \text{max} \rangle$). If the lower constraint is zero then the activity is optional in the process, and may be omitted in an instantiated model. Otherwise, a specialization of the activity must be included. The upper constraint specifies the maximal number of specializations of that activity which are allowed in an instantiated model. The following are common multiplicity indicators and their usage.

- An activity whose multiplicity indicator is $\langle 0, 1 \rangle$ is optional in an instantiated model. However, if it is included, there may be only one possible specialization of it.
- An activity whose multiplicity indicator is $\langle 0, * \rangle$ is optional in an instantiated model. If it is included, it may have many possible specializations.
- An activity whose multiplicity indicator is $\langle 1, 1 \rangle$ is mandatory in an instantiated model, and there is exactly one possible specialization of it.
- An activity whose multiplicity indicator is $\langle 1, * \rangle$ is mandatory in an instantiated model, and there may be many possible specializations of it.

To specify multiplicity indicators, the modeling language used in conjunction with ADOM is required to employ a classification mechanism that can be attached to its elements. Such mechanism exists in some languages (e.g., the stereotype and profile mechanisms in UML). If the language used does not include such mechanism, its syntax should be extended. A similar classification mechanism is used in the instantiated model in order to connect between specialized activities and their generic

counterparts in the reference models. This connection is essential for validation purposes, which is out of the scope of this paper. Note that multiplicity indicators are also attached to flow elements, such as triggers, pre-/ post-conditions, and transitions, in order to constrain the possible flows between the specialized activities.

In Figure 1 we demonstrate ADOM's support for the various specialization operations through a partial example of a Sales process reference model (Figure 1a) and its instantiation in a chocolate manufacturing organization (Figure 1b). The modeling language used is EPC, whose syntax is extended to include a classification mechanism. EPC employs functions as activities and events as an explicit representation of their initial and final sets of states.

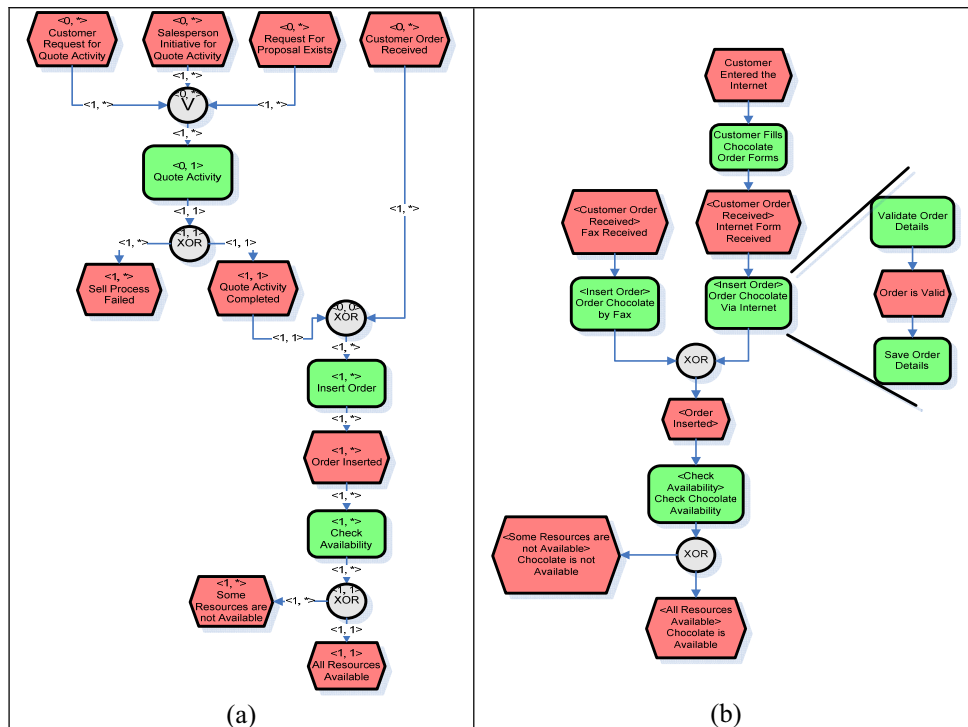


Fig. 1. Part of a Sales process reference model (a) and its instantiation to a chocolate manufacturing organization (b)

The partial reference model includes an optional single *Quote Activity*, which may be triggered by a number of possible events. This activity may end in failure or in success, when it is followed by an *Insert Order* activity, which is a mandatory activity that can be unlimitedly specialized. Alternatively, *Insert Order* may be triggered by *Customer Order Received* events only. In this case, *Quote Activity* and its related events, connectors, and flows (all defined as optional) will not appear in the instantiated model. These two alternative paths to invoke *Insert Order* are decided at design time as specified by the <0, 0> multiplicity indicator of the xor connector. In any case, *Insert Order* is followed by a mandatory *Check Availability* activity, which

can be specialized unlimitedly. Note that the reference model does not refer to, neither constrain, the possible types of specialization operations that can be applied to these activities, thus, any specialization operation is allowed.

Figure 1b, which is an instantiation of the reference model to a chocolate manufacturing organization, demonstrates all the specialization operations mentioned in this paper. To avoid misinterpretation, the activities and events of the instantiated model are classified according to the reference model elements.

The refinement operation is demonstrated by the *Order Chocolate via Internet* activity which is further refined (in a separate EPC) into the activities *Validate Order Details* and *Save Order Details*. As can be seen, the pre-condition of these two sequential activities is *Internet Form Received* which is classified as *Customer Order Received*, and their post condition is *Order Inserted*, as required by the *Insert Order* activity in the reference model. The activities *Order Chocolate by Fax* and *Order Chocolate Via Internet* are sub-types of the reference model *Insert Order* activity. These multiple sub-types are allowed, as the multiplicity indicator of their corresponding activity in the reference model is $\langle 1..* \rangle$. The *Check Availability* activity is specialized by contextual adoption to *Check Chocolate Availability* as it refers to a distinguished subject.

Examining the reference model and its instantiated model, one can observe that the reference model part that deals with *Quote Activity*, including the activity itself and its pre- and post-conditions, was omitted from the instantiated model, as it is not relevant to the chocolate manufacturer. This operation is allowed since the *Quote Activity* and its pre-conditions are classified as optional elements in the reference model¹. The instantiated model also includes a unique part which is relevant only for orders that arrive via the Internet and, hence, does not appear in the generic reference model, *Customer Fills Chocolate Order Forms*.

6 Conclusion

The reuse process of reference models is usually not guided. In particular, such guidance is missing in models that apply reuse by specialization, which is an important type of reuse that allows a high level of flexibility and variability in the instantiated processes. This paper proposes an approach that facilitates reuse by specialization by using a domain engineering approach called ADOM.

ADOM enables representing knowledge gained about business processes together with guidelines for creating new business processes that follow the same (reference) model. The application of ADOM to business process reference models enables a rich expressiveness of the specification of specialization operations. In this paper, we identified five specialization operations, namely refinement, sub-typing, contextual adoption, omission, and inclusion. These operations can be applied with different

¹ Note that some of the elements, namely the arcs and the post-conditions, are marked as mandatory. However, since they are dependent of the *Quote Activity*, they remain redundant after omitting the activity and, hence, are omitted too (following the EPC semantics). Their necessity is only relevant if the *Quote Activity* is specialized.

modeling languages. We exemplified the five specialization operations in EPC, which originally was not geared to reference model specification, but rather to business process modeling. These operations extend the variability options of instantiating business processes, thus may increase the applicability of the generic model.

Future research will develop a structured instantiation process, where specialization operations will form a basis for a set of corresponding specialization patterns. The processes of instantiating business process models and validating them according to reference models is planned to be automated and plugged-into relevant Computer-Aided Process Engineering (CAPE) tools.

References

- [1] Czarnecki, K. and Eisenecker, U. W. *Generative Programming - Methods, Tools, and Applications*, Addison-Wesley, 2000.
- [2] Curran, T. A. and Ladd, A., *SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain Management*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1999.
- [3] Fettke, P., Loos, P. and Zwicker, J. "Business Process Reference Models: Survey and Classification", *Proceedings of First International Workshop on Business Process Reference Models*, Nancy, France, pp. 1-15, 2005.
- [4] Reinhartz-Berger, I., Soffer, P., and Sturm, A., "A Domain Engineering Approach to Specifying and Applying Reference Models", *Proceedings of Enterprise Modeling Information Systems Architecture (EMISA '05)*, Klagenfurt, Austria, 2005.
- [5] Reinhartz-Berger, I. and Sturm, A., "Behavioral Domain Analysis – The Application-based Domain Modeling Approach", the 7th International Conference on the Unified Modeling Language (UML'2004), LNCS 3273, pp. 410-424, 2004.
- [6] Rosemann, M. and Aalst van der, W. M P. "A Configurable Reference Modeling Language", *Information Systems* 32 (1), pp. 1-23, 2007.
- [7] SCOR Reference model, Supply chain council version 8.0. www.supply-chain.org, 2007.
- [8] Scheer, A. W. *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer Berlin, 1998.
- [9] Specialization, WordNet® 2.1. Retrieved February 17, 2007, from Dictionary.com website: <http://dictionary.reference.com/browse/specialization>.
- [10] Sturm, A. and Reinhartz-Berger, I., "Applying the Application-based Domain Modeling Approach to UML Structural Views", the 23rd International Conference on Conceptual Modeling (ER'2004), LNCS 3288, pp. 766-779, 2004.
- [11] Thomas O., "Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation", *Proceedings of First International Workshop on Business Process Reference Models*, Nancy, France pp.16-28, 2005.
- [12] Van Es, R., *Dynamic Enterprise Innovation*, Baan Business Innovation B.V., The Netherlands, 1998.