Extending the Adaptability of Reference Models

Iris Reinhartz-Berger, Pnina Soffer, and Arnon Sturm, *Member, IEEE*

***Abstract*— Reference models are an important aid for business process modeling and design. Their aim is to capture domain knowledge and assist in the design of enterprise specific business processes. The application of reference models for process design requires guidance in reusing these models, as well as flexibility in adapting them to specific enterprises. A popular modeling language for specifying reference models is Event-driven Process Chains (EPC), which has been extended for expressing configurable reference models (C-EPC). These models provide explicit reuse guidance, but allow a limited level of flexibility following a reuse by configuration approach. To increase the level of adaptability of reference models, we propose in this paper to utilize the Application-based Domain Modeling (ADOM) approach for the purpose of specifying and applying reference models using EPC. ADOM supports the enforcement of reference model constraints while allowing a high level of flexibility, adaptability, and variability in the business processes of particular enterprises. The paper presents the syntax and semantics of the proposed approach, called ADOM-EPC, and its specialization and configuration capabilities. ADOM-EPC is evaluated by comparing it to C-EPC, a leading approach for reference modeling and reuse, in terms of expressiveness and comprehensibility. While the expressiveness of ADOM-EPC, namely, its set of specified reuse operations, exceeds that of C-EPC, the understandability of the two types of reference models is similar.**

***Index Terms*—Business Process Design, Domain Analysis, Modeling, Reference Models, Reuse, Systems Analysis and Design**

I. Introduction

Business process modeling and design has gained much attention in recent years. While diversity of business processes among organizations is high, there are many common aspects that apply to the majority of the organizations which share common characteristics (e.g., market segment, size, logistic typology, etc.). This fact has been widely recognized and motivated the emergence of a number of reference models, defined as models used for supporting the construction of business process models [22]. In practice, a reference model is a model that captures common processes, well known and practiced in a variety of organizations, rather than a specific process that exists in some organization. Reference models can be reused in order to assist in process design in specific enterprises. Thus, they can support organizations to be effectively designed for new missions [11]. In a wider sense reference models can be used for system engineering ([8], [19]) and software engineering ([9], [15]).

Manuscript received December 9, 2008.

I. Reinhartz-Berger is with Department of Management Information Systems at Haifa University, Haifa 31905, Israel (Tel: 972 -4- 8288502; Fax: 972 -4- 8288522; e-mail: iris@mis.haifa.ac.il).

P. Soffer is with Department of Management Information Systems at Haifa University, Haifa 31905, Israel (e-mail: spnina@mis.haifa.ac.il).

A. Sturm is with the Department of Information Systems Engineering at Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel (e-mail:sturm@bgu.ac.il).

Different methods have been suggested for specifying reference models. These methods refer to different views of the specified reference models, such as the process view, the data view, and the organizational view. The views aim at creating an integrated knowledge-base of reference models which can be searched for guiding business designers while modeling data, events, processes, and flows related to particular organizations.

While much attention has been given over the years to the construction of reference models and to the knowledge that is captured in them, the process of reusing this knowledge through process design in a specific organization has been somewhat neglected until recently. Lately, a configurable reference modeling approach has emerged, using the Event-driven Process Chains (EPC) notation for specifying reference models. This approach, called Configurable EPC (C-EPC) [16], facilitates configuration of reference models for a specific business process as well as provides guidance (as part of the reference model) for the configuration process. However, the reuse operations allowed by this approach enable a limited flexibility in the business processes to be designed based on the reference model. In particular, there is no support for specializing the model for a specific context [1] or adding enterprise-specific functions within the configured business model. Considering the implementation of enterprise systems, reports (e.g., [4]) show that while a large portion of the newly designed business processes rely on available reference models, some enterprise specific modifications and additions are still needed. Such adaptations go beyond configurations of reference models.

In [14], a domain engineering approach, called Application-based DOmain Modeling (ADOM), was proposed as a platform for reference models. When adopting ADOM to reference models, the reference model resides as an upper-level, specifying and enforcing constraints on the specific business processes. Thus, this approach allows flexibility while providing guidelines and validation templates for utilizing reference models. ADOM can be applied to a variety of modeling languages [15], but when adopting ADOM with a specific modeling language, this language is used for specifying both business processes and reference models, facilitating the tasks of business process design and validation.

The utilization of ADOM for reference models proposed in [14] uses Business Process Modeling Notation (BPMN) and concentrates on validation of particular business processes with respect to reference models.

In this paper we examine the expressiveness and comprehensibility of ADOM in the context of representing and utilizing reference models for business process design. We claim that the ADOM approach better fits this purpose than other reference modeling approaches, most notably C-EPC. In particular, ADOM facilitates reuse of reference models' knowledge in a more flexible and expressive manner than C-EPC, supporting both configuration and specialization operations. Due to the extensive use of EPC in modeling business processes, and particularly in order to benefit from the experience of C-EPC and enrich it, we use EPC as the modeling language in this work. We refer to this dialect of ADOM as ADOM-EPC.

The remainder of the paper is organized as follows. In Section II we present background information about business process reference models and discuss two important types of reuse processes: reuse by configuration and reuse by specialization. For each type of reuse process, we elaborate on its reuse guidance and supported flexibility. Section III introduces ADOM-EPC, referring to its representation capabilities, while Section IV elaborates on the specialization and configuration capabilities of ADOM-EPC. In Section V ADOM-EPC is evaluated by comparing it to C-EPC. Finally, Section VI concludes and refers to future research plans.

II. Background

Business process reference models can be used for a variety of purposes: (1) creating standard business processes that can be used in various organizations, (2) guiding the design of new business processes, (3) enforcing organization policies, (4) training new or junior business designers, (5) improving communication between different business process stakeholders, (6) comparing business processes that can be described by the same reference model, and (7) assisting in the implementation of enterprise systems (these models, often attached to enterprise systems, specify processes that can be adopted by the enterprises when implementing these systems).

Business process reference models have been discussed, classified, and evaluated using a number of evaluation frameworks and criteria (e.g., [6], [7], [13], and [18]). The main focus of these evaluation frameworks is the reference models themselves, while less attention is explicitly given to their intended reuse process. Nevertheless, all reference models are designed considering (at least) an implicit intended reuse process. Two important types of reuse processes in reference models are specialization and configuration. Both types are quite flexible and support associating a variety of business processes to the same reference model. The main difference between these two types is the abstraction level of the reference model. While models intended to be reused by specialization are specified at a high level of abstraction, to be refined when specialized to the specific needs of an organization, models intended to be reused by configuration are detailed and at a low abstraction level. Their reuse process entails selecting a subset of the specified processes to be active in the specific organization and customizing these processes in the new context without changing their abstraction level.

Reuse by configuration, as presented in [16], is specifically targeted at reference models attached to enterprise systems, whose application is part of the implementation of the system in an enterprise. The development of reuse by configuration approaches was motivated by the need to guide the adaptation of reference models to meet the specific needs of organizations. To this end, configuration possibilities as well as their dependencies are explicitly specified in the model, making a clear distinction between design decisions and run-time decisions. Specifying configuration possibilities facilitates the adoption of parts of the detailed model without altering its level of abstraction.

The main representative of this approach is Configurable Event-driven Process Chains (C-EPC). While supporting inclusions or omissions of configurable elements, the adaptability of C-EPC is limited in three main ways. First, it does not support specifying different variants of the same configurable element in the enterprise process. Such support is required, for example, when the enterprise has several ways of performing a task, although the inputs, outputs, sub-functions, and flows of these tasks are the same, or very similar. Second, C-EPC does not support configuring events. Finally, C-EPC does not support introducing new model elements, such as functions and events, that are specific to the organization at hand and thus do not appear in the reference model.

As mentioned above, models whose intended use is through specialization (e.g., [12],and [21]) provide knowledge at a high level of abstraction, allowing a high degree of flexibility and facilitating variability in the specific models to be constructed. The main advantage of reuse by specialization is that the knowledge captured in the reference model serves for constructing the specific model without imposing a detailed solution. However, specialization of a model is not a simple task. Most reference models that follow the reuse by specialization approach (e.g., [12]) do not provide guidance regarding how specialization can be accomplished, and this hampers their usability. Such models can be viewed as advices rather than tools that support process design.

The aim of ADOM-EPC is to support and extend the adaptability of reference models. This means that it should allow a high level of flexibility and variability, while providing concrete guidance to the reuse process. Based on a compact extension of EPC syntax, which can relate to models at various abstraction levels, we provide guidance to both reuse by specialization and reuse by configuration, as presented in the following sections.

III. ADOM-EPC

Reference models capture generic knowledge, and hence they introduce additional challenges that are not supported by a process modeling language alone, such as expressing the allowed variability among business processes within a domain or an organization. When creating a specific model, this variability can be manifested in three kinds of operations: (1) omitting reference model elements which are not relevant for a particular organization, (2) including one or more locally-adapted variants of some reference model elements, and (3) introducing specific elements which are not part of the well-known standards, as specified in the reference model. All these are possible in general, but some restrictions should be made so, e.g., specific essential 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. As explained in Section II, configuration enables only the first operation, inclusion or omission of reference model elements. It does not support the second and the third operations, namely including multiple local variants of an element or newly introduced elements. Hence the variability supported by pure configuration models is limited.

To cope with the aforementioned challenges, we enhanced the EPC syntax with two types of classifiers that are added to all EPC elements, including functions, events, connectors, and arcs. The two types of classifiers are multiplicity indicators and reference model classifiers, which are formally defined in the appendix, along with definitions of reference and business process models (see Definitions 1-5). *Multiplicity indicators* are attached to reference model elements and denote the possible lowest and upper-most numbers of variants these elements may have in a business process model. These are denoted by <min, max> attached to the reference model elements. The default multiplicity <0, *> implies no constraints, thus will not explicitly appear in the reference model. The multiplicity indicators enable the reference model to capture commonality (model parts that are common to many organizations) and variability (possible variants) of the modeled domain. Commonality is expressed as mandatory elements, i.e., elements whose minimal value of multiplicity indicator is equal to or larger than 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, and (3) application specific elements, i.e., elements that appear in the specific business process model and have no counterparts in the reference model.

The *reference model classifiers* are associated with elements in the specialized models (i.e., the specific business process models), and denoted by <reference model element name> attached to the specialized model element. These classifiers 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 *specialized or configured by* the specific (business process) model elements, potentially providing more information about the specific situation. A single reference model element may have more than one specialization in a business process model, specifying different variants of the generic elements (e.g., different ways of receiving customer orders).

The rest of this section explains and demonstrates the representation of reference models (Section A) and their specialized models (Section B) in ADOM-EPC through a Sales process. The specification of that process aggregates knowledge from several reference models ([17], [21], [23]), and the MIT process handbook [12], so, e.g., parts that appear in all the models were considered as mandatory, while parts that appear only in some were considered as optional.

A. Representing Reference Models in ADOM-EPC

Figure 1 depicts a Sales process reference model in ADOM-EPC. According to this reference model, the process can start either with a quote activity or with the reception of a customer order. If the organization employs a *Quote activity* (and, hence, does not receive immediate customer orders), there are three different events that may trigger the process: *Customer Request for Quote Activity*, *Salesperson Initiative for Quote Activity*, and *Request for Proposal Exists*. Furthermore, as implied by the implicit multiplicity indicator <0, *>, a particular enterprise may have several events of each type. In case the quote activity is instantiated in the process, it can terminate in either failure or success. Note that this reference model allows an organization to define several events of failure in the quote activity, but only one event that represents its successful completion.

The reference model also allows several types of order insertion functions, as there might be different ways to insert orders in a specific enterprise. Furthermore, it does not impose an order in which these activities will be executed (i.e., in parallel, sequentially, etc.). After the order is inserted, two optional parallel paths may be taken (i.e., a specific business process model may include both, only one or neither): checking the customer credit and validating the order configuration. Afterwards, the availability of raw materials, products, resources, etc., which are required for fulfilling the order, is checked. This function results either in an event where all the required resources are available or in an event where at least some of them are not. Note that the reference model, which includes only generic enough information, does not specify actions to be taken when some of the resources are not available. Cases that are not specified in the reference model can be added specifically when the particular business process model is created.

Finally, there are at least one *Delivery* function and at least one *Payment* function, which are executed in parallel (or independently). Upon completion of all delivery and payment functions the order is closed and the Sales process successfully completes.

B. Representing Specialized Models in ADOM-EPC

A *specialized model of a reference model* is a business process model that follows the guidelines of the reference model and fulfills its constraints. Any particular element in the business process model can be associated to a reference model element via the reference model classifier, implying that the specific element plays the role of the reference model element in the business process model.

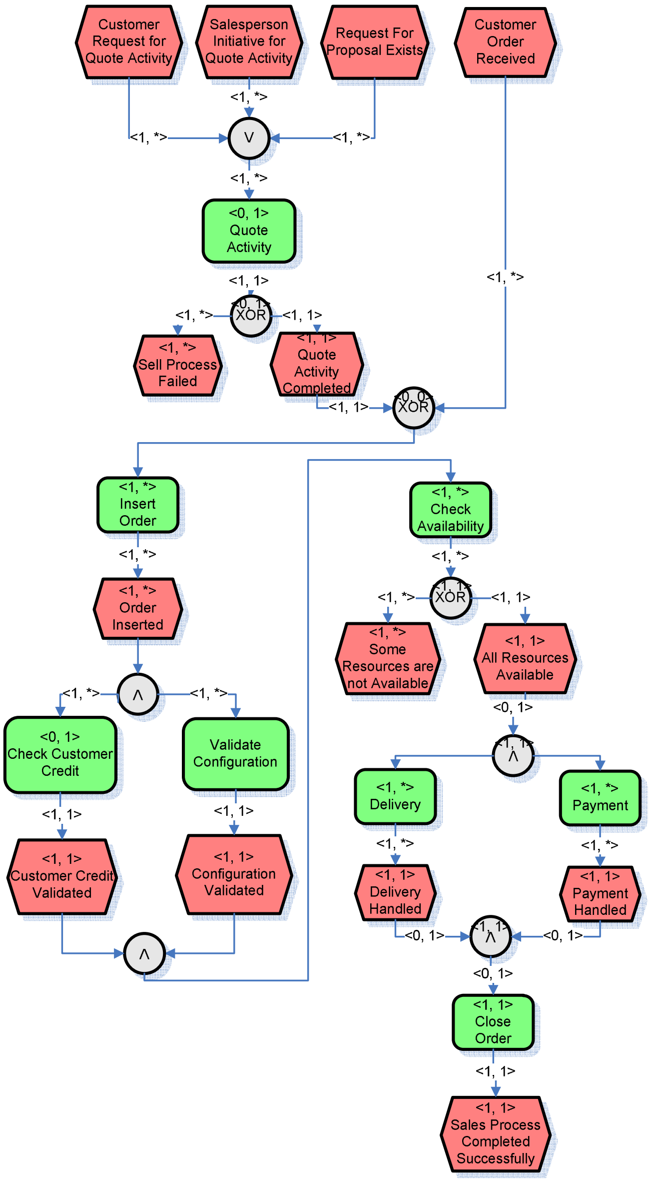


Figure 1. An ADOM-EPC model of the Sales process reference model

Two operations guide the creation of specialized models from reference models in ADOM-EPC: specialization and configuration. *Specialization* is the result of concretization of a reference model element into a specific business process model element. Generic (reference) elements can be specialized through operations of refinement, sub-typing, and contextual adoption [20]. Visually, specialized elements have, in addition to their reference model classifiers, their own (unique) names in the context of the specific business process model.

Configuration is the selection of a subset of existing elements from a reference model for the purpose of specifying a lawful specific business process model. In this case the business process and reference model elements are at the same abstraction level. Usually, configured elements adopt their reference model element names and, hence, visually, only their reference model classifiers appear.

Figure 2 presents a specific Sales process which specializes the Sales process reference model for a chocolate manufacturer that sells various kinds of chocolate from stock. This specific Sales process does not include a quote activity and is triggered by the arrival of a customer order, since the chocolate manufacturer's customers are regular customers working on a long-term basis. Furthermore, a customer order can arrive by phone, by fax, or via the Internet, each way is treated somewhat differently, thus a single reference model function is specialized into several functions according to the needs of the specific enterprise. In contrast, the resulting event of *Order Inserted* is a configured element and needs no specialization in this enterprise. This specific Sales process does not require any *Validate Configuration* function, as it deals with standard products. Hence the (optional) function is discarded, leaving a single path sequence in which the customer credit is checked.

The availability check of the required amount of chocolate and packages may result in one of three cases: all the resources (i.e., both chocolate and the packages) are available, enabling the continuation of the process to delivery and payment; the chocolate is available but the packages are not, requiring the process to wait for the packages to arrive before continuing; and the required amount of chocolate is not available, requiring the cancellation of the order and termination of the process in a failure.

If the process continues, then, prior to delivery, the required transport conditions are checked, since some products need to be delivered in temperatures that do not exceed a certain threshold. This is a function that does not appear in the reference model since it is particular to the enterprise at hand, and is added in the specific process. Note that this addition does not violate any of the constraints imposed by the reference model, including the requirements on the process order, its optional and mandatory elements, etc.

Finally, shipment to the customer may take one of three different paths, all specializations of the reference model *Delivery* function. This is done in parallel to receiving the customer's payment for the order.

IV. Specialization and Configuration in ADOM-EPC

Having introduced ADOM-EPC representation capabilities, in this section we specify the specialization and configuration operations of the approach. The full formalism is provided in the appendix. We start by presenting the extended EPC notation for reference models and specialized models.

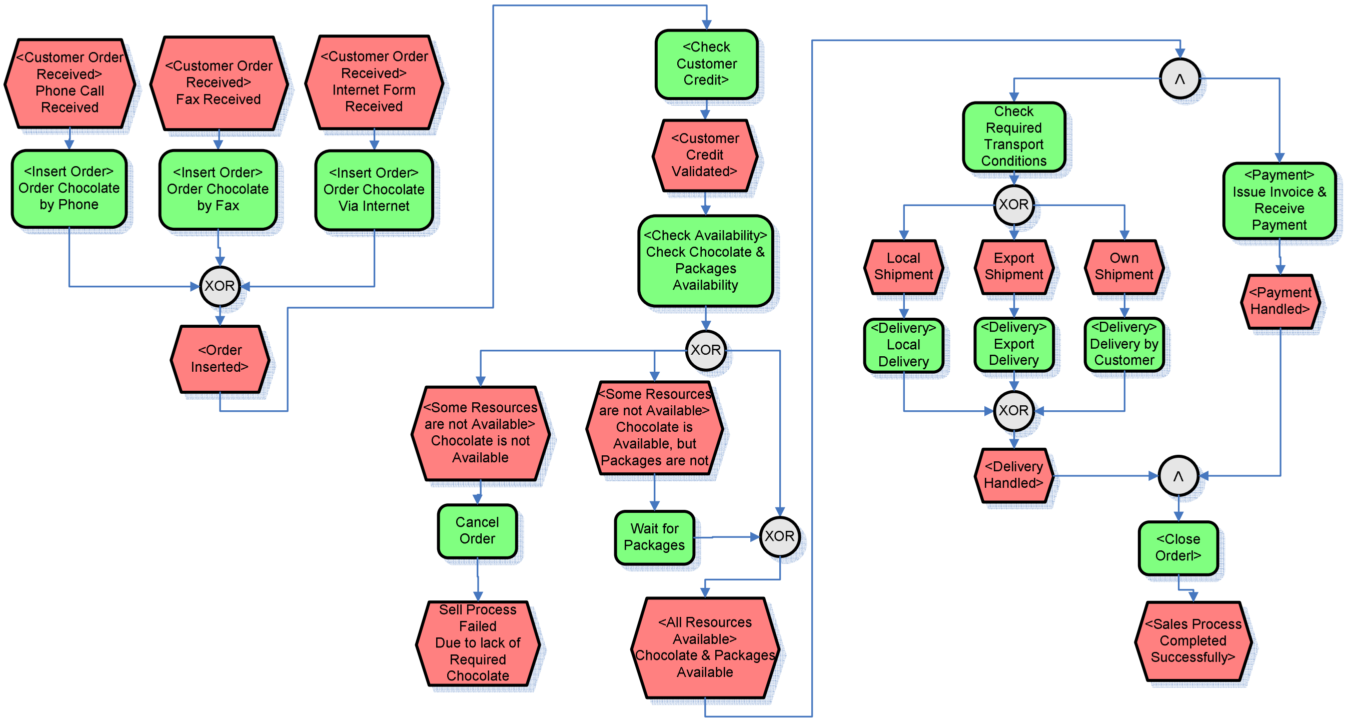


Figure 2. An ADOM-EPC model of the Sales process of a Chocolate Manufacturer

Next, we present the possible specialization and configuration operations of different model elements. Finally, the specialization of the entire model is considered.

The basic elements in EPC are events, functions, connectors, and arcs. Rosemann and van der Aalst [16] define rules for connecting these elements in order to create syntactically correct EPCs (see Definitions 1-3 in the appendix). These rules specify, for example, that the sets of events, functions, and connectors in an EPC are pair wise disjoint, each event in an EPC has at most one incoming arc and at most one outgoing arc, there is at least one start event and at least one final event, events and functions create alternating chains, etc. A syntactically correct EPC may include several different paths, i.e., sequences of nodes (events and functions) that are connected via arcs. These paths are related by connectors which induce logical relationships on the paths. As examples of three paths in Figure 2 consider Phone Call Received → Order Chocolate by Phone → Order Inserted → Check Customer Credit → Customer Credit Validated, Fax Received → Order Chocolate by Fax → Order Inserted → Check Customer Credit → Customer Credit Validated, and Internet Form Received → Order Chocolate via Internet → Order Inserted → Check Customer Credit → Customer Credit Validated. These three paths are connected via a XOR connector, meaning that in any execution of the process exactly one of these paths is traversed.

ADOM-EPC deals with two types of models, both are addressed by our extension of the EPC notation: reference and business process models. A *reference model* is a syntactically correct EPC in which each element (event, function, connector, and arc) is associated with a multiplicity indicator, i.e., a pair of numbers (k, m) where k and m are natural numbers, m ≥ k, and m may be ∞ (represented as * in the diagrams). A *business process model* that corresponds to a reference model is a syntactically correct EPC, in which events and functions can be explicitly associated to events and functions in the reference model. The exact definitions of these models are respectively given in Definitions 4 and 5 in the appendix, whereas Figures 1 and 2 respectively exemplify reference and business process models.

For a business process model to be considered a specialized version of a reference model, it needs to preserve the business logic specified in the reference model, and satisfy its constraints. For defining specialized models, we first define specialization with respect to four types of element: function, event, path (arc), and connector, formalized in Definitions 6-12 in the appendix. Note that since configuration operations are a subset of specialization operations, these definitions are applicable to configuration as well.

A ***business process function f specializes a reference model function f'*** iff the reference model classifier of f is f'. In the Sales process example, the functions *Order Chocolate by Phone*, *Order Chocolate by Fax*, and *Order Chocolate via Internet* in Figure 2 are all specializations of the reference model function *Insert Order* from Figure 1, as indicated by their reference model classifiers.

While functions are addressed as independent elements in the specialization of a reference model, events are addressed in the context of the functions that yield them. An event e in model M *depends* on a function f in the same model iff the path from f to e in M includes only arcs and connectors (and no events or functions). As an event may depend on more than one function, we define the *dependee group of an event* e as all the functions in M such that e depends on them. As an example, the event *Order Inserted* in Figure 2 has three functions in its dependee group: *Order Chocolate by Phone, Order Chocolate by Fax,* and *Order Chocolate via Internet.* The event *Customer Credit Validated* depends only on the function *Check Customer Credit*. Finally, the dependency groups of *Phone Call Received*, *Fax Received*, and *Internet Form Received* are the empty set. Events whose dependency groups are empty are termed *start* or *external events*.

A ***business process event e specializes a reference model event e'*** iff (1) the reference model classifier of e is e', and (2) if e depends on at least one function in the reference model, then e' depends on at least one specialization of that function. In the example of Figure 2, the event *Chocolate is not Available*, whose dependee group includes the function *Check Chocolate and Packages Availability*, specializes the event *Some Resources are not Available* (Figure 1), whose dependee group includes the function *Check Availability*. Note that the function *Check Chocolate and Packages Availability* specializes the reference model function *Check Availability*.

Considering the specialization of paths, we require specialization of its source and destination as well as maintaining the ordering of the specialized elements. A ***business process path p specializes a reference model path p'*** iff (1) the source of p (be it an event or a function) specializes the source of p', (2) the destination of p (be it an event or a function) specializes the destination of p', and (3) for each two directly connected reference model nodes (events or functions) that are specialized in the business process model, the two specializations are connected via a path. Note that since a minimal path consists of one arc connecting two nodes, the path specialization definition is in fact applicable also as an arc specialization.

In the chocolate manufacturer example, all the three paths in Figure 2 between the event *Chocolate & Packages Available* and the event *Delivery Handled* specialize the reference model path in Figure 1 between *All Resources Available* and *Delivery Handled*.

Specialization of EPC connectors relates to their type ($\wedge$, $\vee$, XOR), role (split or join), and context within the process or the reference model. Note that unlike other element types, connectors which represent decision points (namely, XOR and $\vee$) in a reference model may relate to decisions that should be made at design time, at run time, or both. In particular, if the decision is made at design time (when the business process model is constructed), the connector may not appear in the business process model at all or may have a different type. For example, consider the $\vee$ connector between the events *Customer Request for Quote Activity*, *Salesperson Initiative for Quote Activity*, and *Request for Proposal Exists* in Figure 1. In some organizations it may be that only one of these possibilities exists. Then the decision is made at design time, and the specialized model will include only one event and no connector. Other organizations may have more than one possibility. These possibilities may include a XOR relationship among them, denoting that in a specific process only one event can exist (i.e., the decision is made at run time). Alternatively, they may require all events ($\wedge$) or allow two or more events ($\vee$) in the same particular process. All these are valid specializations of the connector. Generally, each reference model connector can be specialized by a business process connector of the same type and role, except for a reference model OR connector, which can be specialized by business process connectors of the same role but different types. This is termed *specialization equivalence* of connectors, as summarized in Table 1 (see also Definition 11 in the appendix).

Table 1: Instantiation equivalence

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

The multiplicity indicators of connectors can force a decision to be made at design time or at run time, or allow the organization to decide on which one to take. A multiplicity indicator of <0, 0> implies that the connector should not appear in a particular model, and the decision it stands for should be made at design time. As an example, consider the XOR connector between the events *Customer Order Received* and *Quote Activity Completed* in Figure 1 whose multiplicity indicator is <0,0>. This implies that a specific model can either include quotation handling or direct order insertions, but both cannot exist as options in the same process. An organization can choose how its process operates when the process is designed. The business process in the chocolate manufacturer case (Figure 2) supports only direct order insertion. In contrast, the XOR connector that follows the function *Check Availability* in Figure 1 has a multiplicity indicator of <1,1>, implying that it must be included in a particular model, since resources may or may not be available every time the process is enacted (run-time decision). Finally, the $\wedge$ connector between the functions *Check Customer Credit* and *Validate Configuration* in Figure 1 has a multiplicity indicator of <0,1>, implying it may or may not exist in a particular model. Since the business process in the chocolate manufacturer case (Figure 2) does not require any configuration validation, this connector is not included in the model.

A ***business process connector c specializes a reference model connector c'*** iff (1) their roles are the same, (2) their types are specialization equivalent, (3) if c is a split connector then it splits at least two specializations of reference model paths split by c' (in the reference model), and (4) if c is a join connector then it joins at least two specializations of reference model paths joined by c' (in the reference model).

Continuing with our example, the ∧ connector between the events *Delivery Handled* and *Payment Handled* in Figure 2 is a specialization of a connector between these functions in Figure 1. The XOR connector above *Delivery Handled* in Figure 2 is not a specialization of any reference model connector.

It is now possible to define specialization of models. A ***business process model specializes a reference model*** iff (see Definition 13 in the appendix for a formal definition):

(1) Each reference model function whose multiplicity indicator is (k, m) has k to m specializations in the business process model.

(2) Each start reference model event (i.e., an external event that is not yielded from a function) whose multiplicity indicator is (k, m) has k to m specializations in the business process model.

(3) For each intermediate or final reference model event whose multiplicity indicator is (k, m) and at least one function in its dependee group is specialized in the business process model, there are k to m specializations of that event related to each specialization of a function in its dependee group.

(4) Each reference model arc whose multiplicity indicator is (k, m) and its source and destination are specialized in the business process model has k to m paths between the specializations of its source and destination.

(5) Each reference model connector whose multiplicity indicator is (k, m) has k to m connector specializations (in the business process model).

The business process model of the chocolate manufacturer depicted in Figure 2 specializes the Sales process reference model presented in Figure 1. As seen in this example, a specialized model can include elements which are not specializations of reference model elements. These are specific additions that uniquely exist in the particular enterprise and are termed *enterprise-specific elements* (see Definition 14 in the appendix). In Figure 2 the functions *Cancel Order*, *Wait for Packages*, and *Check Required Transport Conditions* do not have reference model classifiers and, hence, are enterprise-specific functions. The XOR connector which precedes *Order Inserted* is also enterprise-specific, since it is not a specialization of any reference model connector.

In summary, the operations described above utilize the multiplicity indicators as a clear specification of the boundaries within which a specific model can be constructed on the basis of a reference model. These boundaries capture the business logic embedded in the reference model. The reference model classifiers keep trace of the specialization process and help process designers verify that the reference model constraints are kept. In addition, the formal definitions of the operations (in the appendix) can serve for a formal verification of a specialized model in terms of the business logic specified in the reference model (see [14]).

V. Comparing ADOM-EPC to C-EPC

To evaluate ADOM-EPC, this section compares it to C-EPC, which is the main approach that addresses the reuse process of reference models. In the comparison, we relate to two aspects: expressiveness and comprehensibility. The theoretical comparison, presented in Section A, refers to the expressiveness of the approaches with respect to the guidelines they provide, while the experimental comparison, presented in Section B, refers to the comprehensibility of reference models expressed in the two approaches.

A. Comparison of Expressiveness

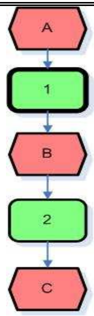
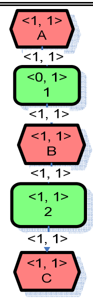
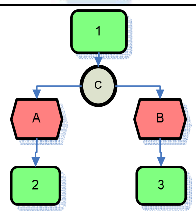
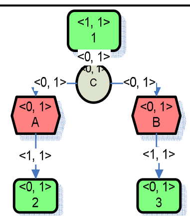
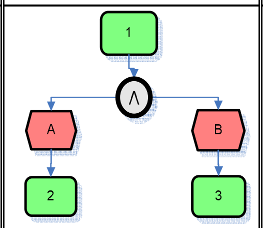
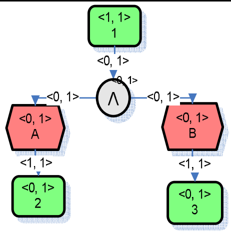
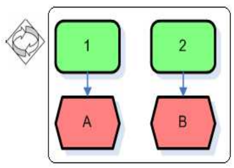
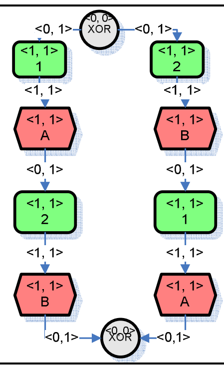
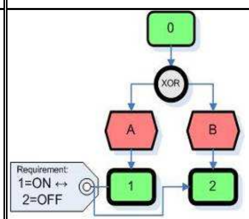
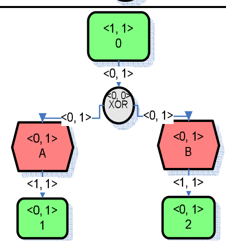
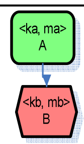
The differences between the two approaches lie in the different perspective over the required reuse operation. While C-EPC refers to configuration only, assuming that organizations should align their business processes with the "best-practice", ADOM-EPC adopts the specialization approach. This approach considers organizations to follow common business logic in their own processes, while the details of the specific process they implement may be different. The ADOM-EPC perspective is also supported by [4], in which a measurement of the level of reuse of the SAP reference models in a number of case studies belonging to different market segments indicates that full reuse was not achieved in any of them, although in some cases the level of reuse was remarkably high. The parts that were not fully reused were either modified or designed from scratch.

Still, ADOM-EPC allows configuration operations as a simplified version of specialization. Table 2 summarizes various patterns for reuse of reference models, indicating their support in ADOM-EPC and C-EPC. Most of the patterns are configuration patterns originated in [5]. Note that we joined the split and join versions of the patterns and the 'OR' and 'XOR' connectors (both denoted as "C" in the table), as their behavior is similar in both approaches. This table shows that all the configuration patterns supported by C-EPC are also supported by ADOM-EPC, although interleaved parallel routing and sequence inter-relationships are only partially supported in ADOM-EPC. Regarding the interleaved parallel routing pattern, ADOM-EPC's representation is less compact than the one of C-EPC and may become overloaded when a large number of sequence combinations exist. ADOM-EPC, on the other hand, allows two reuse operations which are not supported by C-EPC, namely multiple specializations and enterprise additions.

The difference between ADOM-EPC and C-EPC can best be understood by examining the Sales process example shown in Figure 1 and Figure 2. A C-EPC reference model from which the model of Figure 2 can be derived must be a detailed model, including at least all the elements in Figure 2, and many others that may exist in other organizations. It would necessarily include a function of *Check Required Transport Conditions*, although this function is not applicable to most organizations.

Note that many options can also exist for the quotation handling part, which is not even included in the specific model of Figure 2. Yet, the model would not be able to specify all the options that can exist in possible other specializations of the relatively compact ADOM-EPC reference model of Figure 1.

Table 2. ADOM-EPC vs. C-EPC: a theoretical comparison

Pattern	Description	C-EPC	ADOM-EPC	Comments
Optionality	A function in a sequence that can be switched off in an instantiated model			C-EPC allows optional switching ("opt"), explicitly leaving the decision to run time. In that case, two alternative paths should be included in the specific business process: (a) with the execution of function 1 and (b) without its execution. C-EPC does not prescribe explicitly what to do with events A and B if function 1 is discarded, ADOM-EPC relates to design time decisions, so run-time optionality is not related to this pattern, and is allowed anyway. If function 1 is discarded, event B is discarded too.
XOR/OR split / join	A connector whose number of paths can change or it can be reduced to a sequence			C can be either ∨ or XOR. The connector can be specialized according to the specialization equivalence matrix (ADOM-EPC) or configurable connector constraints (C-EPC). Note that the rules implied by both are the same (see [5])
AND split / join	A connector whose number of paths can change			Both C-EPC and ADOM-EPC allow mapping a configurable AND connector to an AND connector. However, ADOM-EPC also allows omitting the AND connector, implying that the specialized business process consists of a single path.
Interleaved parallel routing	The order in which functions appear in the instantiated model should be determined at design time, as long as the functions are executed sequentially			ADOM-EPC explicitly specifies the possible sequences, while enforcing design time decision (by the XOR connector whose multiplicity indicator is <0, 0>). This model becomes complicated as the number of possibilities increases. Note that in order to avoid this, C-EPC introduced a new symbol to EPC.
Sequence inter-relationship	Two inter-related functions, in this case mutually exclusive, so if one is included in the specific model the other should not appear			ADOM-EPC enables expressing only exclusiveness between two functions and does not cover other possible situations supported by C-EPC requirements and guidelines.
Multiple specializations	A function, an event, a path, or a connector can be specialized several times in a business process model	N/A		If ma>1, then A can be specialized more than once in a single business process model, but at most ma times. Furthermore, each specialization of A is connected to between kb and mb specializations of B.
Enterprise additions	An enterprise-specific function, event, path, or connector can be included in a business process model	N/A	Implicitly allowed	This operation is not constrained, and can always be performed in ADOM-EPC. Hence, it is not indicated in the reference model.

B. Experimental Comparison of Comprehensibility

Based on the theoretical comparison and the Sales process example, it seems that ADOM-EPC models are much more compact and better support flexibility than C-EPC models. However, the operations of reuse by configuration which relate to one abstraction level are much simpler than those of reuse by specialization, which relate to two abstraction levels and require a higher cognitive effort. Hence, the usability of ADOM-EPC may be questioned. In order to assess and compare an important aspect of usability of ADOM-EPC and C-EPC, we conducted an experiment whose aim was to analyze the comprehension of reference models expressed in C-EPC and ADOM-EPC for inexperienced or junior business process designers. The hypotheses, settings, and results of the experiment are reported below.

Experiment Hypothesis

Since ADOM-EPC refers to two levels of abstraction, we planned to check whether this has an effect over the comprehension of the reference model and the provided guidelines it encapsulates. In particular, we wanted to check whether the understanding of the possible reuse operations is different in ADOM-EPC as compared to C-EPC. Thus, we formulated the following hypothesis:

H0: The understanding level of a reference model and of its possible reuse operations is the same in ADOM-EPC and in C-EPC.

H1: The understanding level of a reference model and of its possible reuse operations is different in ADOM-EPC and in C-EPC.

Experimental Settings

The subjects of the experiment were fourth year students in a four-year engineering B.Sc. program at Ben-Gurion University of the Negev, Israel, who took the course “Information Systems in Industry” in the year 2008. The students belonged to two study programs, namely, Information Systems Engineering and Software Engineering. They had already worked in industrial projects and had knowledge and experience in modeling and design. Thus, they may be comparable to junior business process designers. During the course, the students studied various aspects of information systems in industry, where several sessions were devoted to business process specifications, reference models, EPC, C-EPC, and ADOM-EPC.

The experiment took place as part of the final examination of the course. That part of the examination included two tasks, in which the students were asked to respond to eight true/false comprehension questions about reference models expressed in C-EPC and ADOM-EPC.

The students were divided arbitrarily into two groups of 20 students each. They were provided with alternating form types according to their seating positions, so this arbitrary division into the two experimental groups closely approximated random division. Each group got a different test form type. In the forms of type A, the first task referred to a purchase requisition management reference model expressed in C-EPC and the second task referred to a purchase order management reference model expressed in ADOM-EPC. In the forms of type B, the order of the tasks as well as their modeling approaches were inversed.

The comprehension questions referred to lawful business processes that can be generated from the provided reference models. These relate to mandatory and optional functions and events, to multiple elements of the same types, to enterprise specific elements, and to relationships among elements[1].

Experiment Results

The comprehension questions were checked according to a pre-defined detailed grading policy, which included potential errors along with the number of points that should be reduced for each error. Each comprehension question could score a maximum of 1 point (8 points in total for each model). Incomplete answers, or incorrect answers, scored less according to the detailed grading policy. The scores were then normalized on a scale of 0-100.

Table 3 summarizes the average scores of the comprehension grades. Since the sample size was relatively small, we could not assume a normal distribution of the scores. Hence, we used the Mann-Whitney non-parametric test to corroborate the hypothesis. The results show that although the average comprehension scores of the C-EPC models were slightly higher than those of the ADOM-EPC models, the difference was not found statistically significant, thus H0 cannot be rejected. Our conclusion is that the understandability of ADOM-EPC models is not different than that of C-EPC models, while its expressive power is higher than that of C-EPC.

Table 3. Results of the comprehension average grades

	ADOM-EPC		**C-EPC**		**z**	**p-value**
	Mean	**Var**	**Mean**	**Var**		
Purchase Requisition	52.5	27	55.31	19	-0.95	0.924
Purchase Order	62.81	12	65.94	26	-1.172	0.241

Although the results may be applied to inexperienced or junior business process designers, as discussed above, we believe that the reuse guidance provided by ADOM-EPC may also be beneficial for expert business process designers as in today’s environment business processes are the back bone of the business and are frequently becoming more complex. However, this hypothesis has to be checked in industry.

VI. CONCLUDING DISCUSSION

Proposing ADOM as a platform for specifying and specializing reference models relies on knowledge existing within the domain engineering discipline. As opposed to

[1] The reference models, the comprehension questions, and the expected answers with respect to both C-EPC and ADOM-EPC approaches can be found at http://mis.hevra.haifa.ac.il/~iris/research/ADOM-EPCexp1.pdf.

existing reference modeling approaches, the proposed ADOM-EPC approach supports both reuse by specialization and configuration of reference models in business process modeling. Thus it overcomes limitations of other approaches, allowing both a high level of adaptability and reuse guidance. The evaluation of the expressiveness of ADOM-EPC shows it is capable of supporting the configuration patterns of C-EPC as well as other patterns which are not supported by C-EPC. This expressiveness reflects the larger set of reuse operations that are supported by ADOM-EPC and enable a broader variability in the specific business process models. This expressiveness is achieved by using a consistent addition to the EPC notation – the multiplicity indicators in the reference models and the reference model classifiers in the particular business process models. Furthermore, the experimental findings reported here show that the broader adaptability support does not harm the understandability of its possible reuse operations.

Considering the possible usages of reference models discussed in Section II, these are well supported by ADOM-EPC. An ADOM-EPC reference model can serve as a standard for a variety of business processes and guidance for the design of new business processes in the same organization or market segment. Such model can be much compact than a C-EPC model and yet capture a higher level of variability. When using an ADOM-EPC reference model for training process designers and enforcing organization policies, the reference model provides high-level principles over which specific processes will be designed and validated. In contrast, a C-EPC model provides detailed "best practices" which support only configuration options. The higher abstraction level of ADOM-EPC models can help stakeholders reach agreement and enable comparison of a wide range of processes. However, it may also raise ambiguities and be difficult to communicate about. Finally, both ADOM-EPC and C-EPC reference models can be used for implementing enterprise systems and specifying configuration options supported by these enterprise systems. C-EPC is better suited for this purpose as it supports a limited set of configuration options. An ADOM-EPC reference model, on the other hand, which can also be specialized, may indicate places where the system's functionality can be extended in adaptation to specific needs of the enterprise.

While the experimental results reported above provide evidence regarding the understandability of ADOM-EPC, this is only a first step in assessing its usability. Further empirical evaluations, possibly involving practitioners and experienced business process designers, will enable a better understanding and possible improvements of ADOM-EPC. In addition, further research is required to establish a comprehensive process for guiding reference model specialization.

APPENDIX: ADOM-EPC FORMALISM

Definition 1 *(Event-driven Process Chain)***:** An Event-driven Process Chain (EPC) is a five-tuple (E, F, C, l, A), where:

- E, F, C are respectively finite sets of events, functions, and connectors
- $l \in C \rightarrow \{\wedge, XOR, \vee\}$ is a function which maps each connector onto a connector type
- $A \subseteq (E \times F) \cup (F \times E) \cup (E \times \{c \in C \mid c \rightarrow \vee \in l \text{ or } c \rightarrow \wedge \in l\}) \cup (C \times E) \cup (F \times C) \cup (C \times F) \cup (C \times C)$ is a set of arcs

Definition 2 *(path)***:** A path p from n_1 to n_k is a sequence $<n_1, \ldots, n_k>$, where k>1, $n_i \in E \cup F \cup C$ (i=1…k) and $(n_i, n_{i+1}) \in A$ (i=1…k−1). We say that n_i (i=1…k) belongs to the path p, n_1 is the source of the path, and n_k is its target. The length of a path is defined as the number of arcs in the path (i.e., k-1).

Useful notations: Let EPC = (E, F, C , l, A) be an Event-driven Process Chain:

- The set of nodes of EPC is defined as Nodes = $E \cup F \cup C$.
- $C_\vee = \{c \in C \mid l(c) = \vee\}$; $C_\wedge = \{c \in C \mid l(c) = \wedge\}$; $C_{XOR} = \{c \in C \mid l(c) = XOR\}$.
- source (a), target (a) are respectively the source and target of an arc $a \in A$
- The set of input and output nodes of a node $n \in$ Nodes are respectively $\bullet n=\{m \mid (m,n)\in A\}$, $n\bullet=\{m \mid (n,m)\in A\}$
- The set of join and split connectors are respectively $C_J = \{c \in C \mid |\bullet c| \geq 2\}$, $C_S = \{c \in C \mid |c\bullet| \geq 2\}$
- $C_{EF} =\{c \in C \mid$ c belongs to a path of length 2 from an event to a function$\}$
- $C_{FE} = \{c \in C \mid$ c belongs to a path of length 2 from a function to an event$\}$
- $C_{EE} = \{c \in C \mid$ c belongs to a path of length 2 from an event to an event$\}$
- $C_{FF} = \{c \in C \mid$ c belongs to a path of length 2 from a function to a function$\}$

Definition 3 *(syntactically correct EPC)***:** An Event-Driven Process Chain EPC = (E, F, C, l, A) is syntactically correct if and only if the following requirements are satisfied:

- The sets E, F, and C are pairwise disjoint, i.e., $E \cap F = \emptyset$, $E \cap C = \emptyset$, and $F \cap C = \emptyset$.
- For each $e \in E$: $|\bullet e| \leq 1$ and $|e\bullet| \leq 1$.
- There is at least one event $e \in E$ such that $|\bullet e| = 0$ (i.e. there is at least one start event).
- There is at least one event $e \in E$ such that $|e\bullet| = 0$ (i.e. there is at least one final event).
- For each $f \in F$: $|\bullet f| = 1$ and $|f\bullet| = 1$.
- For each $c \in C$: $|\bullet c| \geq 1$ and $|c\bullet| \geq 1$. Furthermore, $|\bullet c| = 1 \rightarrow |c\bullet| > 1$ and $|c\bullet| = 1 \rightarrow |\bullet c| > 1$.
- C_J and C_S induce a partition of C, i.e., $C_J \cap C_S = \emptyset$ and $C_J \cup C_S = C$.
- C_{EE} and C_{FF} are empty, i.e., $C_{EE} = \emptyset$ and $C_{FF} = \emptyset$.
- C_{EF} and C_{FE} induce a partition of C, i.e., $C_{EF} \cap C_{FE} = \emptyset$ and $C_{EF} \cup C_{FE} = C$.

Definition 4 *(Reference model)***:** A reference model in ADOM-EPC is a seven-tuple (E, F, C, l, A, Mult, *mi*), where:

- (E, F, C, l, A) denote a syntactically correct EPC.

- $MULT \subseteq N \times (N \cup \{*\})$ is a set of multiplicity pairs (where N is the set of the natural numbers and * represents ∞). Furthermore, $\forall (k, m) \in MULT$, $m \geq k$ must hold. The elements in MULT are termed *multiplicity indicators*.
- *mi*:$EL \rightarrow MULT$ is a function, where $EL=E \cup F \cup C \cup A$ is the set of elements in the EPC. $\forall n \in E \cup F \cup A$, *mi*(n) = (k, m) satisfies $m>0$[2]. min(el) returns the lowest multiplicity of the element and max(el) returns the upper-most multiplicity of the element, i.e., *mi*(el)=(min(el), max(el)).

Definition 5 *(Business process model)***:** A business process model in ADOM-EPC that corresponds to a reference model RM = (E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) is a seven-tuple (E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*), where:

- (E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}) denotes a syntactically correct EPC.
- RC is a set of elements in the reference model RM, $RC \subseteq EL^{RM}=(E^{RM} \cup F^{RM} \cup C^{RM} \cup A^{RM})$.
- *cl*:$EL^{BP} \rightarrow RC$ is a mapping, where $EL^{BP}=E^{BP} \cup F^{BP} \cup C^{BP} \cup A^{BP}$[3]. The elements in RC are termed *reference model classifiers*.

Definition 6 *(function specialization)***:** Let BP= (E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*) be a business process model and RM= (E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) – a reference model. A business process function $f^{BP} \in F^{BP}$ *specializes* a reference model function $f^{RM} \in F^{RM}$ iff *cl*(f^{BP})=f^{RM}.

Definition 7 *(dependency)*: An event e^M in model M[4] *depends* on a function f^M in the same model iff the path from f^M to e^M includes only arcs and connectors. Notation: $e^M \lrcorner f^M$.

Definition 8 *(dependee group)***:** The *dependee group of* e^M is defined as $D_e{}^M=\{f^M \mid e^M \lrcorner f^M\}$.

Definition 9 (*event specialization*)**:** Let BP= (E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*) be a business process model and RM= (E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) – a reference model. A business process event $e^{BP} \in E^{BP}$ *specializes* a reference model event $e^{RM} \in E^{RM}$ iff:

(1) *cl*(e^{BP})=e^{RM}.

(2) If $D_e{}^{RM} \neq \emptyset$, then $D_e{}^{BP} \neq \emptyset$ and $\exists f^{BP} \in D_e{}^{BP}$, $f^{RM} \in D_e{}^{RM}$ such that f^{BP} instantiates f^{RM}. In other words, if the event depends on at least one function in the reference model, then its specialization depends on at least one specialization of that function.

Definition 10 *(path specialization)*: *A path p^{BP} from a source element s^{BP} to a target element t^{BP} in a business process model* BP=(E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*) *specializes a path p^{RM} from s^{RM} to t^{RM} in a reference model* RM=(E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) iff:

(1) s^{BP} specializes $s^{RM} \wedge t^{BP}$ instantiates t^{RM}.

(2) For each arc $a^{RM}=(n_i{}^{RM}, n_{i+1}{}^{RM}) \in p^{RM}$ such that there exist n^{BP}, $n'^{BP} \in p^{BP}$, n^{BP} specializes $n_i{}^{RM} \wedge n'^{BP}$ specializes $n_{i+1}{}^{RM} \Rightarrow$ there exist a path from n^{BP} to n'^{BP} that is included in p^{BP}.

Definition 11 *(specialization equivalent)***:** A type t of a connector is *specialization equivalent* to a type t' if a connector c^{BP}, *l*(c^{BP})=t, can be considered as a specialization of a connector c^{RM}, *l*(c^{RM})=t'. The cell $SE_{t\ t'}$ in Table 1 defines whether a type t is specialization equivalent to a type t'.

Definition 12 *(connector specialization)*: *A connector c^{BP} in a business process model* BP=(E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*) *specializes a connector c^{RM} in a reference model* RM=(E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) iff:

(1) Their roles are the same, namely ($c^{RM} \in C_J{}^{RM}$ iff $c^{BP} \in C_J{}^{BP}$) and ($c^{RM} \in C_S{}^{RM}$ iff $c^{BP} \in C_S{}^{BP}$).

(2) Their types are specialization equivalent, i.e., *l*(c^{BP}) is specialization equivalent to *l*(c^{RM}).

(3) If c^{BP} is a split connector than it splits at least two specializations of reference model paths split by c^{RM} in the reference model. Formally expressed, if $c^{RM} \in C_S{}^{RM}$ (and consequently $c^{BP} \in C_S{}^{BP}$), then there exist paths $p_1{}^{BP}$, $p_2{}^{BP}$ in BP and $p_1{}^{RM}$, $p_2{}^{RM}$ in RM, such that:
   a. $p_1{}^{BP} \neq p_2{}^{BP} \wedge p_1{}^{RM} \neq p_2{}^{RM}$
   b. c^{BP} is the source of $p_1{}^{BP}$ and of $p_2{}^{BP}$
   c. c^{RM} is the source of $p_1{}^{RM}$ and of $p_2{}^{RM}$
   d. $p_1{}^{BP}$ specializes $p_1{}^{RM}$ and p_{BP2} specializes $p_2{}^{RM}$
   e. $\neg\exists$a connector $c'^{BP} \in p_1{}^{BP} \cap p_2{}^{BP}$ such that c'^{BP} specializes c^{RM} (meaning that there are no sub-paths that are valid specializations)

(4) If c^{BP} is a join connector than it joins at least two specializations of reference model paths joined by c^{RM} in the reference model. Formally expressed, if $c^{RM} \in C_J{}^{RM}$ (and consequently $c^{BP} \in C_J{}^{BP}$), then there exist paths $p_1{}^{BP}$, $p_2{}^{BP}$ in BP and $p_1{}^{RM}$, $p_2{}^{RM}$ in RM, such that:
   a. $p_1{}^{BP} \neq p_2{}^{BP} \wedge p_1{}^{RM} \neq p_2{}^{RM}$
   b. c^{BP} is the target of $p_1{}^{BP}$ and of $p_2{}^{BP}$
   c. c^{RM} is the target of $p_1{}^{RM}$ and of $p_2{}^{RM}$
   d. $p_1{}^{BP}$ specializes $p_1{}^{RM}$ and p_{BP2} specializes $p_2{}^{RM}$
   e. $\neg\exists$a connector $c'^{BP} \in p_1{}^{BP} \cap p_2{}^{BP}$ such that c'^{BP} specializes c^{RM} (meaning that there are no sub-paths that are valid specializations)

Definition 13 *(model specialization)*: *A business process model* BP=(E^{BP}, F^{BP}, C^{BP}, l^{BP}, A^{BP}, RC, *cl*) *specializes a reference model* RM=(E^{RM}, F^{RM}, C^{RM}, l^{RM}, A^{RM}, MULT, *mi*) iff:

(1) $\forall$function $f^{RM} \in F^{RM}$ such that mi(f^{RM})=(k, m) $\Rightarrow \exists$r functions $fi^{BP} \in F^{BP}$ (i=1, ..., r where $k \leq r \leq m$) such that fi^{BP} specializes f^{RM}.

(2) $\forall$event $e^{RM} \in E^{RM}$ such that *mi*(e^{RM})=(k, m):
   a. If $D_e{}^{RM} \neq \emptyset$ (i.e., e^{RM} is an intermediate or final event), then $\forall f^{BP} \in F^{BP}$ such that f^{BP} specializes $f^{RM} \in D_e{}^{RM}$, $\exists$r events $e_i{}^{BP} \in E^{BP}$ (i=1, ..., r where $k \leq r \leq m$) such that $e_i{}^{BP}$ specializes e^{RM}.

[2] Note that for $c \in C$ *mi*(c)=(k, m) allows $m \geq 0$. The meaning of *mi*(c)=(0, 0) is explained later.

[3] Note that *cl* is not necessarily a complete mapping, i.e., there might be an element $e \in EL^{BP}$ such that *cl*(e) is not defined, implying that e is an enterprise-specific element.

[4] M can be a reference model or a business process model.

b. If $D_e^{RM}=\emptyset$ (namely, e^{RM} is a start event), then $\exists$r events $e_i^{BP} \in E^{BP}$ (i=1, ..., r where $k \le r \le m$) such that e_i^{BP} specializes e^{RM}.

(3) $\forall$arc $a^{RM}=(n_1^{RM},n_2^{RM})\in A^{RM}$, $mi(a^{RM})=(k, m)$, if n_1^{RM} and n_2^{RM} are specialized in BP, then $\exists$r paths p_i^{BP} (i=1, ..., r where $k \le r \le m$) from n_{1i}^{BP} to n_{2i}^{BP} where p_i^{BP} specializes $a^{RM} \wedge n_{1i}^{BP}$ specializes $n_1^{RM} \wedge n_{2i}^{BP}$ specializes n_2^{RM}.

(4) $\forall$connector $c^{RM}\in C^{RM}$ such that $mi(c^{RM})=(k, m)$ $\exists$r connectors $c_i^{BP} \in C^{BP}$ (i=1, ..., r where $k \le r \le m$), c_i^{BP} specializes c^{RM}.

Note the possibility that k=0, in which case it is possible that r=0.

Definition 14 *(enterprise-specific element)***:** Let RM be a reference model and BP – a business process that specializes RM. An *enterprise-specific element* is an element in BP which cannot be considered as a specialization of a reference model element in RM.

References

[1] Becker, J. Delfmann, P. and Knackstedt, R. (2007) Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models, In Becker, J and Delfmann, P. (eds) Reference Modeling: Efficient Information Systems Design Through Reuse of Information Models, Physica-Verlag HD, p. 27-58.

[2] Bernus P. (1999) Generalised Enterprise Reference Architecture and Methodology Version 1.6.3. IFIPIFAC Task Force on Architectures for Enterprise Integration.

[3] Curran, T. and Ladd, A. (1999) SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain Management, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ.

[4] Daneva, M. (2000) Reuse Measurement in the ERP Requirements Engineering Process. In Proceedings of Software Reuse: Advances in Software Reusability, 6th International Conference, Lecture Notes in Computer Science 1844, pp. 211-230

[5] Dreiling, A., Rosemann, M., van der Aalst, W.M.P., Sadiq, W., and Khan, S. (2005) Model-driven process configuration of enterprise systems. In Proceedings of Wirtschaftsinformatik 2005, eEconomy, eGovernment, eSociety (Ferstl, O.K., Sinz, E.J., Eckert, S., and Isselorst, T. Eds), p. 687, Physica-Verlag.

[6] Fettke. P. and Loos, P. (2003) Multiperspective Evaluation of Reference Models – Towards a Framework. In Proceedings of ER'03 Workshops, LNCS 2814 (Jeusfeld, M. and Pastor, O. Eds), p 80, Springer-Verlag, Berlin.

[7] Fettke. P., Loos, P., and Zwicker, J. (2005) Business Process Reference Models: Survey and Classification. In Proceedings of First International Workshop on Business Process Reference Models, Nancy, France, p. 1.

[8] Gehin, A.L. and Staroswiecki, M. (2008) Reconfiguration Analysis Using Generic Component Models. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 38 (3), 575-583.

[9] Graves, A.R. and Czarnecki, C. (2000) Design Patterns for Behavior-Based Robotics. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 30 (1), 36-41.

[10] Lang, K., Glunde, J., and Bodendorf, F. (1997) A framework for reusable reference process building blocks. ACM SIGGROUP Bulletin 18 (1), 68 – 70.

[11] Levchuk, G.M., Levchuk, Y. N., Luo, J., Pattipati, K.R., and Kleinman, D. L. (2002) Normative Design of Organizations—Part I: Mission Planning. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 32 (3), 346-359.

[12] Malone, T.W., Crowston, K., and Lee, J. (1999) Tools for Inventing Organizations: Towards a Handbook of Organizational Processes. Management Science 45 (3), 425-443.

[13] Misic, V.B. and Zhao, J.L. (2000) Evaluating the Quality of Reference Models. In Proceedings of ER 2000 – 19th Conference on Conceptual Modeling, p 484, Springer-Verlag, Berlin.

[14] Reinhartz-Berger, I., Soffer, P., and Sturm, A. (2009) Organizational Reference Models: Supporting an Adequate Design of Local Business Processes. International Journal of Business Process Integration and Management 4 (2), 134-149.

[15] Reinhartz-Berger, I. and Sturm, A. (2007) Enhancing UML Models: A Domain Analysis Approach. Journal on Database Management (JDM) 19 (1), special issue on UML Topics, 74-94.

[16] Rosemann, M. and van der Aalst, W.M.P. (2007). A Configurable Reference Modeling Language, Information Systems 32 (1), 1-23.

[17] Scheer, A.W. (1998). Business Process Engineering: Reference Models for Industrial Enterprises, Springer Berlin.

[18] Schuette, R. and Rotthowe, T. (1998) The Guidelines of Modeling – an Approach to Enhance the Quality in Information Models. In Proceedings of ER 1998 – 17th International Conference on Conceptual Modeling, p 240, Springer, Berlin.

[19] Shenhar, A.J. and Bonen, Z. (1997) The New Taxonomy of Systems: Toward an Adaptive Systems Engineering Framework. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 27 (2), 137-145.

[20] Soffer, P., Reinhartz-Berger, I., and Sturm, A. Facilitating Reuse by Specialization of Reference Models for Business Process Design. CAiSE'07 workshop proceedings, Tapir Academic Press, pp. 339-347, (2007)

[21] 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 3 (4), 471-476.

[22] Thomas, O. (2005) Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation. In Proceedings of First International Workshop on Business Process Reference Models, Nancy, France, p. 16.

[23] Van Es, R. (1998) Dynamic Enterprise Innovation. Baan Business Innovation B.V. The Netherlands.

Iris Reinhartz-Berger is a faculty member at the management information systems department, University of Haifa. She received her B.Sc. in computer science and mathematics in 1994, M.Sc. in information systems engineering in 1999, and Ph.D. in information systems engineering in 2003, all from the Technion, Israel Institute of Technology, Haifa, Israel. Her research interests include conceptual modeling, modeling languages and techniques for analysis and design, domain analysis, development processes, and methodologies.

Pnina Soffer is a faculty member at the management information systems department, University of Haifa. She received her BSc in 1991, her MSc in 1994 and PhD in 2002 from the Faculty of Industrial Engineering and Management at the Technion, Israel Institute of Technology. Between 1994 and 1998 she worked in an industrial company, dealing with production planning and ERP implementation. Her research interests are business processes and conceptual modeling, including ontological foundations of both.

Arnon Sturm (M'09) received his B.Sc. in industrial engineering and management in 1995, M.Sc. in management and operations research in 1999, Ph.D. in Information System Engineering in 2004, all from the Technion, Israel Institute of Technology, Haifa, Israel.

Between 1995 and 1998 he worked with Israel Defense Force in integrating real-time and information systems. He then moved to Amdocs serving as a team leader. After starting his Ph.D. studies he joined IBM Haifa research labs till 2002. In 2005 he joined Department of Information Systems Engineering at Ben-Gurion University of the Negev. His research interests consist of software development methods, domain engineering, and agent-oriented software engineering.