



Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Dreiling, Alexander, Rosemann, Michael, Van Der Aalst, Wil, & Sadiq, Wasim (2008) From conceptual process models to running systems: A holistic approach for the configuration of enterprise system processes. *Decision Support Systems*, 45(2), pp. 189-207.

This file was downloaded from: <http://eprints.qut.edu.au/30946/>

© Elsevier

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

<http://dx.doi.org/10.1016/j.dss.2007.02.007>

From Conceptual Process Models to Running Systems: A Holistic Approach for the Configuration of Enterprise System Processes*

Alexander Dreiling^{1,4}, Michael Rosemann², Wil M. P. van der Aalst³, Wasim Sadiq⁴

¹European Research Center for Information Systems, University of Münster
Leonardo-Campus 3, 48149 Münster, Germany
alexander.dreiling@ercis.de

²Faculty of Information Technology, Queensland University of Technology
Level 5 / 126 Margaret St, Brisbane QLD 4000, Australia
m.rosemann@qut.edu.au

³Department of Technology Management, Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
w.m.p.v.d.aalst@tm.tue.nl

⁴SAP Research CEC Brisbane, SAP Australia Pty Ltd.
Level 12 / 133 Mary St, Brisbane QLD 4000, Australia
alexander.dreiling@sap.com, wasim.sadiq@sap.com

Abstract

This paper proposes a method which aims at increasing the efficiency of Enterprise System implementations. First, we argue that existing process modeling languages that feature different degrees of abstraction for different user groups exist and are used for different purposes which makes it necessary to integrate them. We describe how to do this using the meta models of the involved languages. Second, we argue that an integrated process model based on the integrated meta model needs to be configurable and elaborate on the enabling mechanisms. We introduce a business example using SAP modeling techniques to illustrate the proposed method.

Keywords: Information Modeling Concepts, Business Process Modeling, Reference Modeling, Model Configuration, System Configuration

* This work has been funded by SAP Australia Pty Ltd. and Queensland University of Technology, Brisbane, Australia and conducted collaboratively between these two institutions in the project “Modeling Configurable Business Processes”.

1. Introduction

The common presupposition of Enterprise Systems (ES) is that they support organizations in their operations and lead to significant efficiency gains. However, this is only true for well-implemented ES that support an organization's processes. The list of major ES project failures is long with famous examples such as FoxMeyer Drug who were allegedly driven into bankruptcy by the implementation of an ES and sued SAP for it [39]. Other examples include Mobil Europe and Dow Chemical both of which spent hundreds of millions of US\$ for their ES implementations [15].

The difficulties arise from the gap between the generic character of an ES and the non-generic, individual character of an organization. Within academia this development is reflected by a constantly growing body of literature on configuration [5, 11, 15, 20, 23, 31, 38] emphasizing that information systems are typically not implemented in an organizational context but adapted to organizational needs from 'off-the-shelf' packages. Major ES vendors similarly aim at tackling this problem by including an increasing amount of adaptation mechanisms into their packages in order to support the configuration process within organizations.

This paper introduces a method which targets increased usability of conceptual modeling for the purpose of ES configuration as conceptual modeling is underutilized in this context [43]. One of the reasons for this is that modeling is often seen to be a tool for documentation purposes only and as such not perceived as a value-adding tool within an ES project. Also, if modeling is used for requirements engineering purposes, usually the models do not automatically impact on the software configuration which again drives the perception that modeling is an overhead. Since modeling is underutilized the question arises as to how to create an improved value proposition related to conceptual modeling as part of an ES project. This paper's approach to achieve this goal features three different aspects:

- Various perspectives of modeling: Managers and technical project members have a different perspective on a business process. To meet the requirements of different user groups, alternative modeling languages have evolved. Changing established modeling languages is time-consuming and may result in resistance of project members to use modeling. We therefore propose to integrate existing process modeling languages.
- Model configuration: A set of predefined conceptual models needs to be adapted to the specific requirements of an organization.
- ES configuration by means of model configuration: Usually, ES software needs to be adapted to the specific requirements of an organization. Graphical, intuitive means for configuring a system, i.e., system configuration by model configuration, are only rare as of today.

The model integration we propose differs from integration concepts underlying techniques such as UML or ARIS. We propose to integrate process modeling techniques which have evolved for different stakeholders such as management or technical analysts. The next section of the paper will elaborate on this topic. Second, we propose to make the integrated languages configurable which will be the concern of the remainder of our paper. Section 3 will discuss the vertical integration of process models subsequently followed by a discussion of process configuration in general in Section 4. Section 5 contains a business example that will provide a better understanding of our approach. Finally, a short outlook will be given and future prospects will be discussed.

2. Perspectives in Process Modeling

Within the fields of Information Systems and Computer Science, numerous process modeling languages have evolved. These techniques vary in their degree of comprehensibility to certain user groups, i.e., they are of different pragmatic quality [30]. Some process modeling languages depict business processes from a high-level perspective with a focus on understanding key points of the process (for instance SAP's Collaborative Business Scenarios). In these cases an intuitive comprehensiveness for a large number of users with typically limited modeling experiences is more

important than a high expressive power and detailed descriptions of a process. Other modeling techniques describe a business process with the purpose of executing the process automatically (workflow languages). Such techniques demand high rigor and express details, but are often only used by a limited number of experienced modelers.

We distinguish between the three perspectives: management, business process analyst, and technical analyst, and discuss them in more detail. This framework follows the commonly accepted distinction between managerial and non-managerial work on the one hand [47], and between business and IT on the other hand. Furthermore, this framework appropriately reflects the nature of ES projects, where the main involved parties are an organization's management, functional departments, its IT department, and external implementation partners. The management perspective allows for communicating process models as description of ES processes and functionality to management. The business process analyst perspective enables implementation partners and organizational actors from functional departments to communicate about the aspired functionality of the ES. Finally, the technical analyst perspective allows for specifying business process in a format that can be processed by IT. These models can be used as a communication tool between IT departments and implementation partners. The remainder of this section will be concerned with discussing these three perspectives in more detail and motivating why they are important in the context of ES configuration.

2.1 Management Perspective

The management perspective on a business process needs to provide a quick and intuitive overview of the business processes of an organization including related inter-organizational business processes. Management is responsible for many or even the entire set of business processes. Therefore, managers have to be aware of how business processes are executed within an organization, in detail. Nevertheless, due to the significant amount of business processes and their complexity, especially in large organizations, it is unavoidable to maintain a certain level of abstraction within the management perspective.

Business process frameworks as the highest level of an enterprise-wide model, provide a glance at the entire set of business processes within an organization. They provide an intuitive starting point for studying the entirety of an organization's business processes. Several of these frameworks have been developed as entry points into rich reference models. For instance, the CIM-Y framework developed by Scheer is comprised of business processes for manufacturers with the two main processes order management and product lifecycle management [37]. The Retail-H, as another example, depicts the business processes involved in retailing [10, 32]. The H-shaped framework includes all processes from procurement, over warehousing, to sales for operatively conducting retail. As a final example, the enhanced Telecom Operations Map (eTOM) is a business process framework of an 'ideally operating' telecommunication company [44].

Several ES software providers have included modeling techniques for process frameworks into their products. SAP, for example, currently provides so-called Solution Maps (SM) and Collaborative Business Scenarios (CBS) as reference models for their support of certain industries such as Automotive, Chemicals, or Retail or cross-industry concepts such as Customer Relationship Management, Supply Chain Management, or Enterprise Resource Planning.

2.2 Business Process Analyst Perspective

The business process analyst perspective is located 'between' the rather high-level management perspective and the detailed perspective of a technical analyst. Unlike the two other perspectives, the business process analyst faces a variety of purposes when it comes to modeling. This includes business process documentation, process improvement, risk management, or knowledge management, as well as software selection, software configuration, system requirements specification, or process simulation. Consequently, this perspective demands rich and adaptable meta models. The notation of these models must be intuitively enough to support interaction with business users, who maybe modeling novices. At the same time, it must feature a degree of rigor, so that these models can form the starting point of a system or workflow development lifecycle. Several modeling languages have

been developed to address the needs of this perspective. For instance, Event-driven Process Chains (EPCs) as an integrating modeling language of the process perspective within the ARIS (Architecture of Integrated Information Systems) approach [36] can be used to express business processes. EPCs have become common as software vendors such as SAP and Siebel have used them for their application-specific reference models.

2.3 Technical Analyst Perspective

The technical analyst perspective focuses on the IT-support of business processes. Within this perspective, it is especially important to represent the parts of business processes that are supported by process-aware information systems such as workflow management systems.

Although workflow management (WFM) has been researched for a significant period now, with many software products available, there is no commonly accepted standard of a workflow language. Modeling standards such as BPEL4WS (Business Process Execution Language for Web Services) or BPML (Business Process Modeling Language) or notations such as BPMN (Business Process Modeling Notation) have been driven by the demand for solutions based on Web services. These standards, however, have a low level of maturity and still lack a significant uptake in practice. Recent publications and research on workflow management in general [6] or on workflow languages in particular [3] based on a rigorous analysis of workflow language requirements [4] suggests that this domain will change significantly over the next years.

Several organizations proposed workflow standards such as the Workflow Management Coalition (WFMC) [48], the RosettaNet Consortium [34], or the Supply Chain Council [42] towards workflow architectures, languages, or specific process schemas. One of the more advanced and influential workflow approaches is the so-called YAWL (Yet Another Workflow Language) approach [3, 19], an open source workflow management system (www.yawl-system.com). This workflow language is based on thoroughly research Workflow Patterns [4]. Especially at an instance level, YAWL features several sophisticated modeling constructs to handle splits and synchronization of workflow branches.

3. Integration of Perspectives

The existence of different process modeling languages in practice confronts an organization with two alternatives:

1. The organization can pursue the ‘conversion’ of all language users within the organization to one conceptual process modeling language.
2. The organization can accept the multiplicity of conceptual process modeling languages and build upon it.

We argue that the first option is not realistic for two main reasons: (1) it is impossible if the target language (i.e. the language to ‘convert’ to) is incapable of expressing constructs that some users have expressed in the past and that is necessary for them; (2) it is furthermore cumbersome or impracticable to achieve a single-language environment if entire language communities (such as product developers, or managers) must ‘convert,’ because they may simply resist these changes. It is plausible to assume that product developers cannot or are not willing to ‘convert’ from a rich workflow language to a high-level conceptual process modeling language that may simply be unsuitable for their purposes. It is furthermore not meaningful to confront managers having a broad area of responsibilities with detailed workflow models in order to inform them about internal processes. Hence, we propose to tolerate the co-existence of different languages and to integrate languages that express business processes at different levels of granularity. The question that arises is, of course, how this integration can be achieved?

In order to provide conceptual support for the implementation of process-aware information systems, language integration of the introduced perspectives needs to be achieved, which requires a mapping of language constructs within the perspectives as shown in Fig. 1. This framework will be populated in Section 5 of this paper when the actual integration of Collaborative Business Scenarios and Event Driven Process Chains will be explained in a business example.

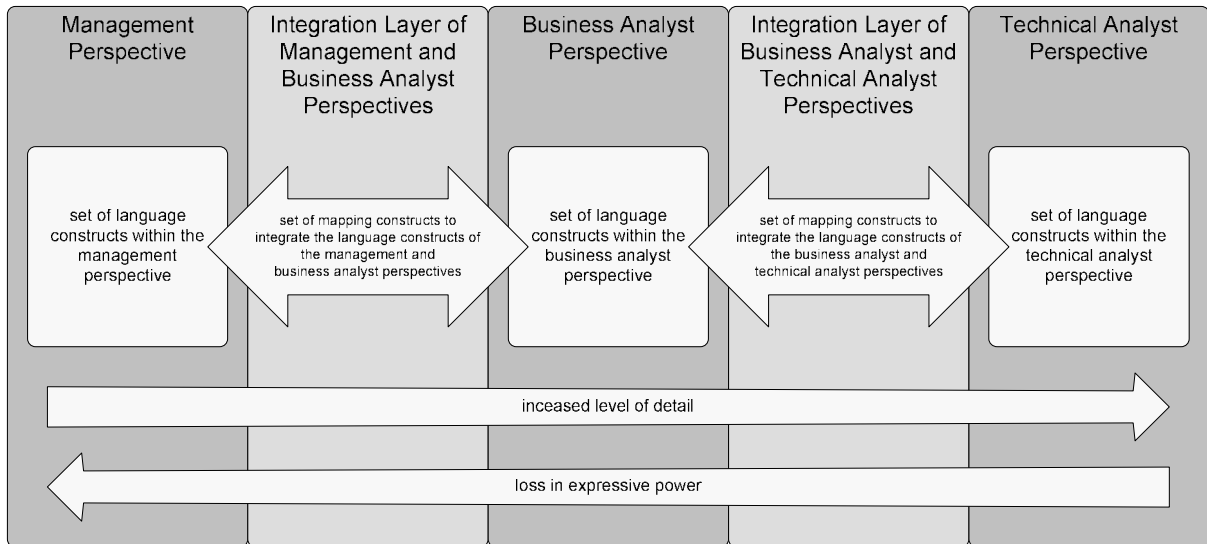


Fig. 1: Generic language integration for the three perspectives Management, Business Process Analyst, and Technical Analyst

Apart from the fact that all ES project members can continue to communicate with each other in the ways they are used to, one of the major advantages gained by an integration of languages of the three introduced perspectives is the impact of configuration decisions between the perspectives. If a top-down approach is chosen within an ES configuration project, e.g. switching-off an activity within the management perspective will allow for switching-off entire processes or process branches within the business process analyst perspective and entire workflow schemas or parts of workflow schemas within the technical analyst perspective automatically. Bottom-up, integrated configuration allows for feedback-mechanisms. If, e.g., a business process analyst discovers problems with the enactment of a business process after a certain process branch has been switched-off due to a configuration decision within the management perspective, he may feed back this information during the next milestone meeting to the management which potentially impacts on the original decision.

Integrating languages featuring a different level of abstraction is naturally bound to losing information towards the more abstract information modeling languages. The difficulties of mapping constructs of languages that feature a different level of abstraction immediately become evident after an ontological examination of the languages. The Bunge-Wand-Weber (BWW) Ontology [45, 46] provides a useful framework for such an ontological evaluation. Constructs from different languages belonging to the

same ontological category (e.g. the BWW constructs *Thing*, *Transformation*, or *State*) can be mapped relatively easy at meta level. This task requires an examination of all language constructs of both languages towards their fit into the classes of the used ontology. If all constructs of both languages have been assigned to exactly on ontological construct, pairs of constructs (mappings) of both languages within one class can be constructed. As an example the language construct *Function* of Event-driven Process Chains (EPC) can be mapped to the language construct *Activity* in a Collaborative Business Scenario (CBS) as both language constructs can be assigned to the BWW construct *Transformation*.

This mapping will usually lead to one-to-many relationships between the statements made in the languages rather than one-to-one relationships. In a detailed EPC model one CBS *Activity* will usually be represented by many EPC *Functions*. Tab. 1 shows a simplified ontological analysis of SMs and CBSs within the management perspective, EPCs within the business process analyst perspective, and YAWL within the technical analyst perspective. This analysis depicts what ontological classes are supported by language constructs of the examined languages at the management, business process analyst, and technical analyst levels.

	SM, CBS	EPC*	YAWL
Thing	x	x	x
Property	-	x	x
State	-	x	x
Transformation	x	x	x
Stable State	-	-	-

Tab. 1: Evaluation of modeling languages with the base constructs of the Bunge-Wand-Weber Ontology [45]. *a detailed ontological analysis of EPCs can be found in [21].

The ontological evaluation of the modeling languages is especially useful as it reveals the incapability of ‘high-level’ languages to make statements about aspects that can only be formulated with more complex languages (e.g., EPCs and YAWL). E.g., the ontological concept *State* is not supported in the management perspective, if SMs and CBSs are used. The main reason for this ontological incompleteness is the deliberately limited meta model of those languages. If the more detailed modeling language features an ontological construct which is not supported in a less detailed

modeling language the abstraction of the business process to the less detailed language will be at cost of losing expressive power. In these cases, mapping at a meta level allows for assigning a language construct of the ontological construct State to another one at management level as one-to-one or one-to-many. If, for instance, the Event within EPCs is mapped to a CBS *Activity* at meta level, all Events within an EPC must be assigned to an *Activity* in a CBS. This is a clear change in the statement embedded within the EPC Events, but enables to switch-off EPC Functions and Events if a CBS activity is switched-off within the management perspective.

Apart from the difficulties that arise from mapping constructs which belong to different ontological classes, a closer examination of the constitutional part of processes—their control flow—and the capability of process modeling languages to depict various aspects of control flow reveals that integrating process modeling languages for different perspectives is a non-trivial task. An evaluation of the introduced languages towards their support of Workflow Patterns (Tab. 2) shows that more abstract process modeling languages insufficiently support control flow.

SAP's Solution Maps as the most abstract of the examined languages are completely incapable of expressing control flow. The purpose of this modeling technique is to provide a very coarse granular picture of what processes are to be supported by an ES. It is mainly used in the business blueprint phase of an ES project to decide which processes to support and for which processes no support is pursued. Control flow as a constitutional part of a process modeling language is not necessary within this language. Collaborative Business Scenarios include document flows rather than control flow. However, the document flow suggests that a function needs to have produced an output document serving as the input document for the next function, which somehow implies an order of activities. It can thus be argued that control flow is implicitly supported to a certain extent. Nevertheless, as shown in Tab. 2, the possibilities of CBS in terms of depicting control flow are limited. As in the case of Solution Maps, control flow is not necessarily the main focus of this modeling language.

At the business process analyst level, the possibilities of expressing control flow increase significantly. EPC support ten of the twenty Workflow Patterns and hence six more than SMs and CBSs combined. This level of detail is required in order to communicate business processes from a business perspective. At this level, more than 1,200 comparatively large EPCs used to describe the capabilities of SAP's product landscape up to SAP R/3 release 4.6c. This is certainly too much to serve as a tool for quickly providing an overview of SAP's capabilities, in a way that SMs or CBSs are intended to do. However, this level of detail is still not sufficient at the technical analyst level for two main reasons. First, in every case where any of the ten not supported Workflow Patterns is needed to describe the control flow of a certain business process, the expressional limits of EPCs are reached. Second, some of the constructs of EPCs are semantically difficult [1], which means that it remains unclear what they express. YAWL's capabilities transcend that of EPCs by supporting all Workflow Patterns with the intentional exception of one (*Implicit Termination* – a pattern which is typically not desirable while describing a workflow) and by being unambiguous about the meaning of its language constructs.

Workflow Pattern	SM, CBS	EPC*	YAWL*
Sequence	x/-**	x	x
Parallel Split	x/-**	x	x
Synchronization	x/-**	x	x
Exclusive Choice	-	x	x
Simple Merge	-	x	x
Multi-Choice	-	x	x
Synchronizing Merge	-	x	x
Multi-Merge	-	-	x
Discriminator	-	-	x
Arbitrary Cycles	-	x	x
Implicit Termination	x/-**	x	-
Multiple Instances Without Synchronization	-	-	x
Multiple Instances With a Priori Design Time Knowledge	-	x	x
Multiple Instances With a Priori Runtime Knowledge	-	-	x
Multiple Instances Without a Priori Runtime Knowledge	-	-	x
Deferred Choice	-	-	x
Interleaved Parallel Routing	-	-	x

Workflow Pattern	SM, CBS	EPC*	YAWL*
Milestone	-	-	X
Cancel Activity	-	-	X
Cancel Case	-	-	X

Tab. 2: Evaluation of modelling languages with Workflow Patterns. *Source: a detailed analysis of the supported Workflow Patterns of EPCs and YAWL can be found in [3]. **Control Flow in CBSs is rather depicted as document flow which technically means, that these Workflow Patterns are not supported. However, the document flow suggests that a function needs to have produced an output document serving as the input document for the next function, which somehow implies an order of activities.

The ontological class of each language construct and its corresponding Workflow Pattern determines which constructs can be mapped to each other while vertically integrating process modeling languages. Technically, SMs, CBSs, EPCs, and YAWL are integrated at meta level, since the meta models of the included languages are comprised of these language constructs and relationships between them.

Fig. 2 shows which models and modeling languages are assigned to what level of abstraction related to a segment of the ‘real world.’ If a model M1 expresses a part of the ‘real world’, meta model M2 is a model of language L1 that is used to express this part of the ‘real world’ [24 – 26, 29, 40]. This language-based meta model [40] is itself expressed in a language L2. The example of Event-Driven Process Chains depicts this relationship: a ‘real-world’ process (part of the ‘real world’) must be placed at the instance level. Its corresponding Event-Driven Process Chain diagram is an abstract model of this process (M1) and expressed in Event-Driven Process Chain notation (L1). M2 is an abstract model of L1 and to be placed at meta level. We will use Chen’s entity-relationship diagrams (ERD) [12] for expressing meta models such as M2, i.e., models of modeling languages. L2 will thus be a Chen-ERD. Whereas M1 contains the actual content of a process (e.g., functions such as ‘place order’ or events such as ‘order placed’), M2 contains modeling constructs (e.g., an entity type for ‘function’ and an entity type for ‘event’). Correspondingly, an instantiated metadata repository of the model M2 (language-based meta model of the actual business process) contains entities that represent functions, events, and other parts of EPCs. In other words, this metadata repository contains all information about process models. Implying that M2 contains an entity type *function* for expressing the EPC notation, the corresponding metadata repository contains a relation *function* that itself

contains items such as ‘place order.’ A manipulation of this metadata repository can be conceived of as one of several configuration mechanisms [8, 9] and will be examined in more detail in the context of configuring vertically integrated process models below.

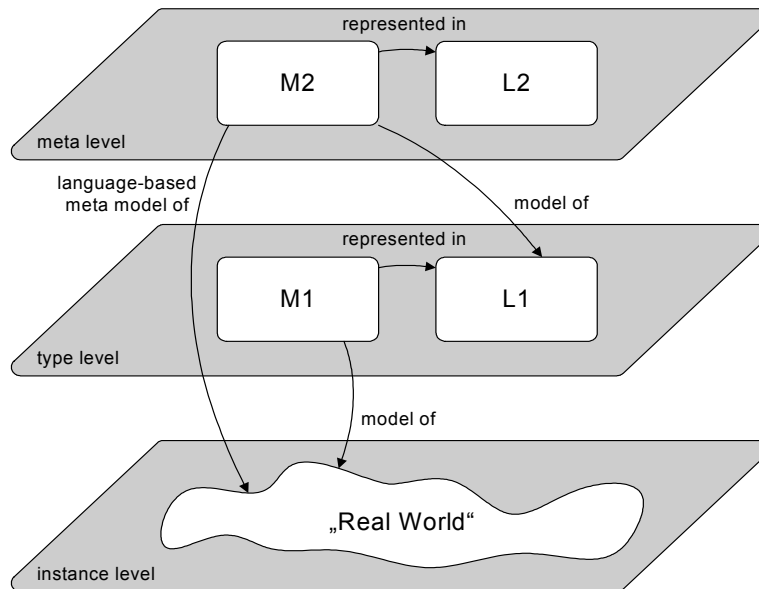


Fig. 2: Models and Modeling Languages on different Levels of Abstraction [24 – 26, 29, 40]

4. Process Configuration

In order to continue this discussion we must make a few remarks on how we actually understand process configuration as the central topic of this contribution.

Configuration of software in order to meet requirements of organizations has been subject to academic discussion for a significant period of time as early examples suggest [20, 31]. Davenport [15] describes the process of configuration as a methodology performed to allow a business to balance their IT functionality with the requirements of their business. More specifically, Soffer et al. [38] describe configuration as an alignment process of adapting the enterprise system to the needs of the enterprise. Especially, if an organization achieved competitive advantages in enacting a business process in a certain way, they usually will not wish to change this business process in order to fit into an enterprise system. In this case, the reference process within the enterprise system needs to be changed according to the ‘real-world’ business process. Soffer et al.’s approach [38] allows for

implementing process variants based on the values of certain attributes. Enterprise system configuration involves setting all the usage options available in the package to reflect organizational features [15]. Brehm et al. [11] define nine different change options for enterprise systems from predefined alterations (e.g. by marking checkboxes) within the enterprise system to alterations of the program code. Holland and Light [23] argue that a critical success factor of enterprise system implementation is to avoid program code changes and wherever possible using predefined change options. Similarly, Sumner and Hamilton [41] argue that minimizing ES configuration significantly contributes to achieve turnarounds in escalated projects. This point clearly hints at insufficient means for ES configuration. In terms of model configuration Becker et al.'s approach is one of the most advanced [9]. It features several mechanisms for transforming a reference model into a build time model. Becker et al.'s approach is very generic and differs from our research in that we, first, seek generic patterns that arise during model configuration and, second, that we propose a configurable modeling language with the CEPC.

Configuration and customization are often used interchangeably. Merriam-Webster's Collegiate Dictionary [33] defines configuration as the "relative arrangement of parts or elements" whereas customizing is defined as "to build, fit, or alter according to individual specifications". With these definitions in mind we can only perform reconfiguration (alteration of relative arrangement of parts or elements within enterprise systems) or customization (alteration of enterprise systems in order to meet the specification of the enterprise). The latter includes alterations of program code, which we do not pursue in our research. We are rather concerned with the configuration of ES and more specifically with that of ES processes. For the purpose of this paper, we define (re-)configuration of an Enterprise System as the process of aligning business aspects such as functions, information, processes, or organization with generic enterprise systems in order to meet the business requirements of the enterprise in the most efficient way. For the sake of simplicity we use the term configuration instead of reconfiguration in this paper.

All of our configuration mechanisms are anchored at meta level. More specifically, we achieve configuration by manipulating a metadata repository that is an instantiation of the meta model M2 in Fig. 2. This metadata repository can easily be built as a relational database, because M2 is an entity-relationship diagram and therefore well-suited for this purpose. Using the example of Event-Driven Process Chains, the metadata repository contains relations (tables in a relational database) for its language constructs such as *Function* or *Event*. Consequently, we are not deleting, e.g., a function within a process model but an entry within the ‘function’ table of a metadata repository. This configuration affects the type level (i.e. the model will be configured) because the information stored within the metadata repository is essentially information about the models. Thus, the basic configuration operators are the following:

- Accept: confirms a preconfigured model / part of the model (does not make changes to a specific part of the metadata repository)
- Delete: an object is removed from the reference model during configuration (an entity is removed from the metadata repository)

These basic operators allow for the construction of more complex operators, out of which we provide a few examples:

- Refine: deletes an object from a model and adds more than one object into which the original object is to be refined to the model (deletes an entry in the metadata repository and replaces it with several others)
- Unify: deletes a number of objects to be unified and adds one unified object to the model (deletes a set of entries in the metadata repository and replaces it with a new one)
- Change: deletes an object and adds another one to the model which represents an alternative to the original object (updates an entry in the metadata repository)

5. Business Example

In order to illustrate the proposed configuration approach, we will now provide a short business example from the domain of Supply Chain Management (SCM). SCM focuses on design, operation,

and maintenance of integrated value chains and aims at satisfying customer needs while simultaneously maximizing customer service [7, 13, 22]. Vendor Managed Inventory has been recently discussed as a concept to increase supply chain efficiency and found especially useful in reacting to volatile changes in demand [17]. For our discussion we will use an SAP example outlined in Fig. 3.

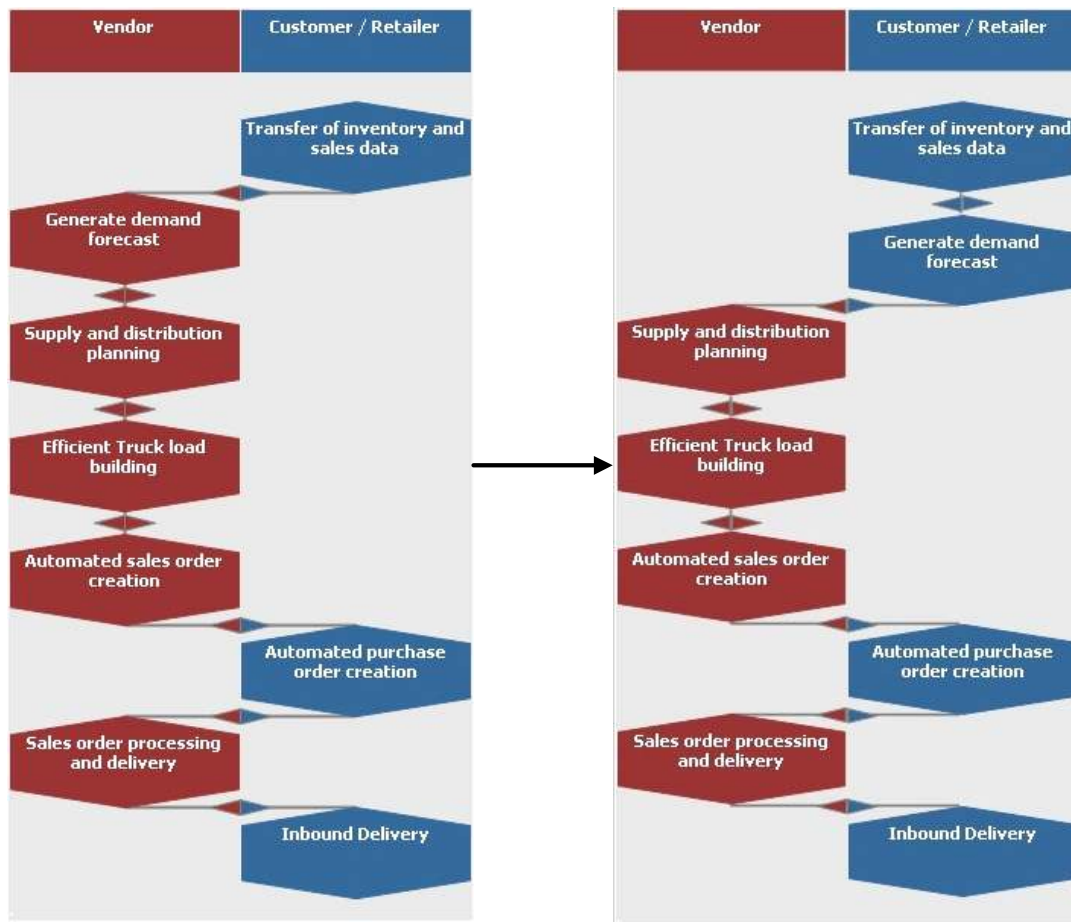


Fig. 3: Configuration of SAP’s Collaborative Business Scenario “Vendor Managed Inventory”. Source (left side): [35]

The example includes an SAP Collaborative Business Scenario which is to be located within the management perspective. It shows in a very abstract way how SAP supports VMI by default. In this example the customer transfers inventory and sales data to respective vendors. In turn, vendors generate demand forecasts and plan supply and distribution of goods. They furthermore allocate trucks for deliveries and automatically generate sales orders. Customers respond with the automatic

creation of corresponding purchase orders which triggers sales order processing on the vendor's side and delivery. The delivery is then received from the customer.

In our example, a company facing volatile demands wishes to engage in Vendor Managed Inventory. However, the demand forecast (CBS activity *Generate Demand Forecast*) should not be done by the vendor as described by SAP's reference model [35], but should have been done in-house, because it is perceived as a competitive advantage by the company. The scenario requires for changing SAP's Collaborative Business Scenario (left side of Fig. 3) into a company-specific model (right side of Fig. 3).

The relevant segment from the vertically integrated meta model to capture the information about language aspects of Collaborative Business Scenarios is introduced in ERM-notation in Fig. 4. A CBS is comprised of (1 to n) CBS Objects (CBSO). Each object may occur in many CBSs. CBSO can be specialized disjoint and equivocally in Activity (A) and Organizational Unit (OU). Activities are performed by organizational units. Document Flow connects two activities with each other denoting that an activity produces an output document which serves as an input document for the next activity.

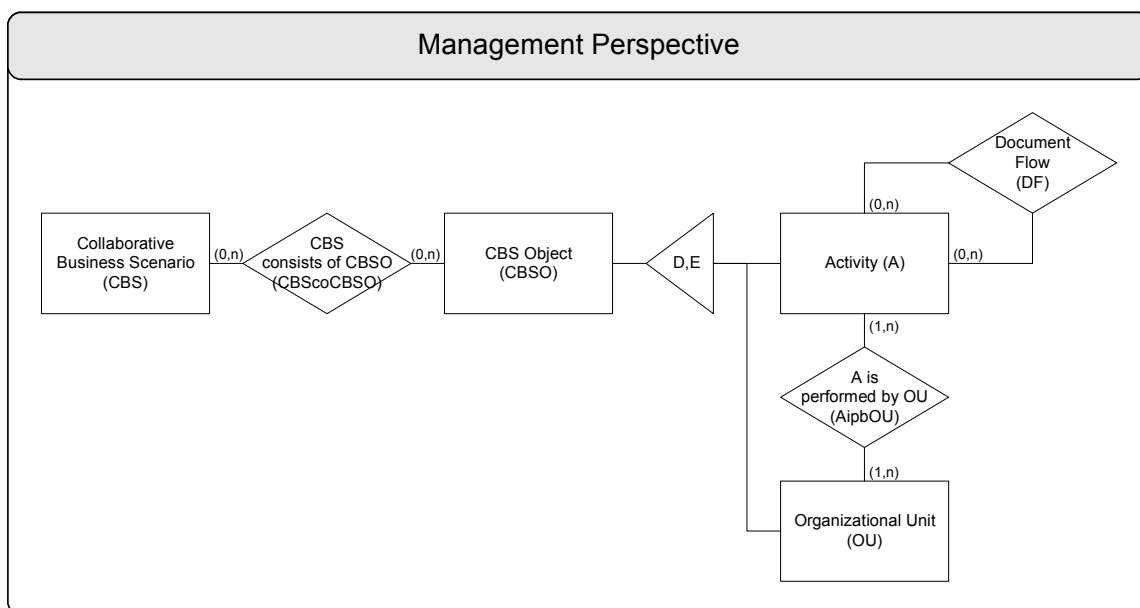


Fig. 4: Segment of the management perspective of the vertically integrated meta model

In order to perform configuration, as described above, we need to define relations for this segment of the meta model. These could be used to create a physical relational database schema. We use theoretically well-founded [12, 14] relational algebra expressions in order to depict how configuration can be achieved using an instantiated metadata repository. The following seven relations (R1-R7) can be defined according to the meta model segment from Fig. 4 (we have underlined key attributes and abstracted from attributes which would be necessary in a real setting such as time frame, cost, etc.). For each entity type we have to introduce one relation (R1, R2, R4, R5). Additionally, we have to introduce a relation for each relationship type (R3, R6, R7) as all relationship types represent n:m relationships.

- R1: $CBS = (\underline{cbsID}, cbsName, cbsVersion)$
R2: $CBSO = (\underline{cbsoID}, cbsoName, objectType, cbsoVersion)$
R3: $CBScoCBSO = (\underline{cbscocbsoID}, cbscocbsoName, cbscocbsoVersion)$
R4: $A = (\underline{aID}, cbsoID, aName, aVersion)$
R5: $OU = (\underline{ouID}, cbsID, ouName, ouVersion)$
R6: $AipbOU = (\underline{aipbouID}, aID, ouID, aipbouName, aipbouVersion)$
R7: $DF = (\underline{dfID}, previousAID, subsequentAID, dfName, document, dfVersion)$

The configuration example introduced in Fig. 3 requires for updating the values of several elements that are included within these relations. We need to query the affected elements (Q1-Q3 expressed in relational algebra) and update the elements derived from these queries (U1-U3). In each query, we assume that a natural join will be done by attributes with exactly the same identifier.

- Q1: Retrieve available information about the activities affected by the configuration:
 $\pi_{\underline{aID}, cbsoID, aName, aVersion} (\sigma_{aName="Generate Demand Forecast"} (A) \bowtie CBSO \bowtie CBScoCBSO \bowtie$
 $\sigma_{cbsName="Vendor Managed Inventory"} (CBS))$
- U1: Update $aVersion$ for the derived set of elements (Changes the element from a reference element into a configured element)

Q2: Retrieve available information about the document flows affected by the configuration:

$$\pi_{dfID, previousAID, subsequentAID, dfName, document, dfVersion} (\sigma_{aName="Generate Demand Forecast"}(A) \triangleright \triangleleft DF \triangleright \triangleleft CBSO \triangleright \triangleleft CBScoCBSO \triangleright \triangleleft \sigma_{cbsName="Vendor Managed Inventory"}(CBS))$$

U2: update *document* for the derived set of elements if necessary (changes the documents exchanged between the activities if the configuration requires it), update *dfVersion* for the derived set of elements (Changes the element from a reference element into a configured element)

Q3: Retrieve available information about activity - organizational unit association (AipbOU):

$$\pi_{aID, cbsoid, aName, aVersion} (\sigma_{aName="Generate Demand Forecast"}(A) \triangleright \triangleleft CBSO \triangleright \triangleleft CBScoCBSO \triangleright \triangleleft \sigma_{cbsName="Vendor Managed Inventory"}(CBS))$$

U3: Update *ouID* for the derived set of elements (Changes the association from the activity “Create Demand Forecast” from the Organizational Unit “Customer / Retailer” to “Vendor”), update *aipbouVersion* for the derived set of elements (Changes the element from a reference element into a configured element)

The ‘version’ attributes in each relation are important to keep track of changes. Consequently, Q1-Q3 and U1-U3 need to be extended by other queries and updating operations that change the ‘version’ attributes of the remaining four relations affected by the configuration if we assume that the right part of Fig. 3 is the final configuration result. Their construction is similar to Q1-Q3 and we omit their discussion here for simplicity reasons.

Configuration so far affected the management perspective solely. Without vertical integration as proposed in the previous section the configuration steps would have to be performed again within the business process analyst perspective which quickly leads to a large overhead of modeling especially within large-scale requirements engineering projects. Moreover, the absence of vertical integration translates into an absence of model-inherent mechanisms to rigorously implement configuration decisions done at management level. The upper part of Fig. 5 thus depicts a segment of the populated framework from Fig. 1 integrating the management and business process analyst perspectives.

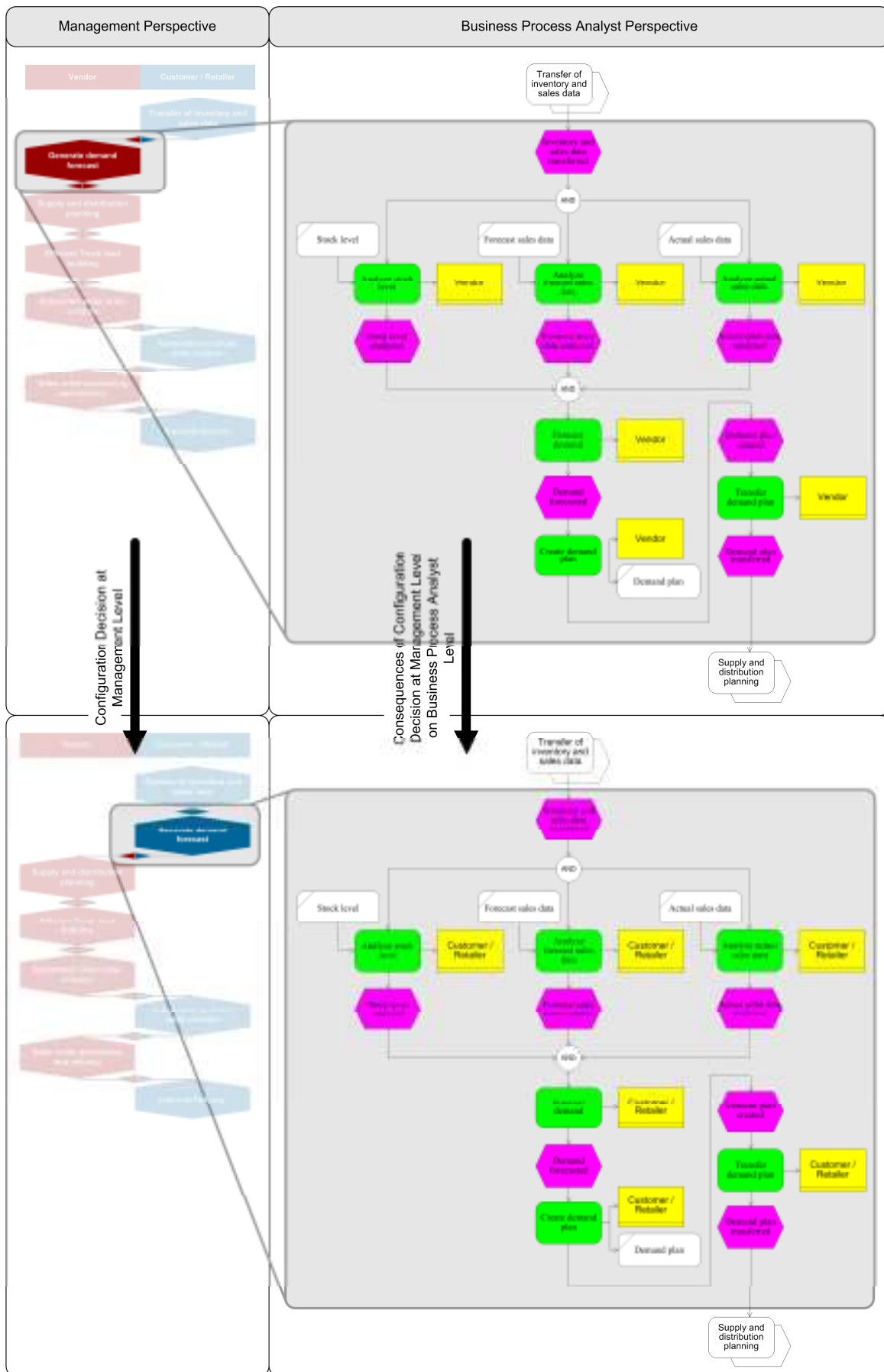


Fig. 5: The configuration decision at management and business process analyst levels

At management level, one CBS activity is concerned with generating the demand forecast (left upper part of Fig. 5). At business process analyst level a comparatively large Event Driven Process Chain corresponds to this activity only (right upper part of Fig. 5). It basically depicts what steps are necessary in order to transform the incoming documents for this activity into the necessary outgoing documents. In the discussion of our business example we already outlined the management decision to shift this activity to the customer side. If the models at management and business process analyst levels were integrated, then the configuration decision at management level can automatically translate into configuration decisions at business process analyst level. This is shown in the lower part of Fig. 5. At this stage, the changes at business process analyst level are changes in organizational responsibility only. This makes sense from a business process analyst perspective because the actual process is performed equivalently by another party. But we will see in the next integration step (between business process analyst and technical analyst) that these changes can as well be more far reaching.

In order to retrieve the information that is affected at business process analyst level by the configuration decision at management level, we need to integrate both languages at meta level as discussed above. The right column of Fig. 6 therefore contains a possible meta model of Event-Driven Process Chains.

Processes are comprised of *Process Objects*, which are specialized into *Connectors*, *Events*, *Functions*, and *Organizational Units*. Each specialized *Process Object* is linked to the others corresponding to the syntactic rules of EPCs. The middle column consists of a vertical integration layer, which links *Collaborative Business Scenario* to *Process* (i.e., an EPC). The many-to-many-relationship accounts for the fact that a CBS will typically be refined by a set of EPCs, but that in turn some EPCs also transcend the processes outlined by a CBS. The CBS construct *Activity* is linked to the EPC construct *Process Object*. Again, the many-to-many-relationship implies less rigor and allows for distinctively different views at the set of organizational activities by

management and business process analysts. A third integration point consists of the link between the CBS construct *Organizational Unit* and the EPC construct *Organizational Entity*.

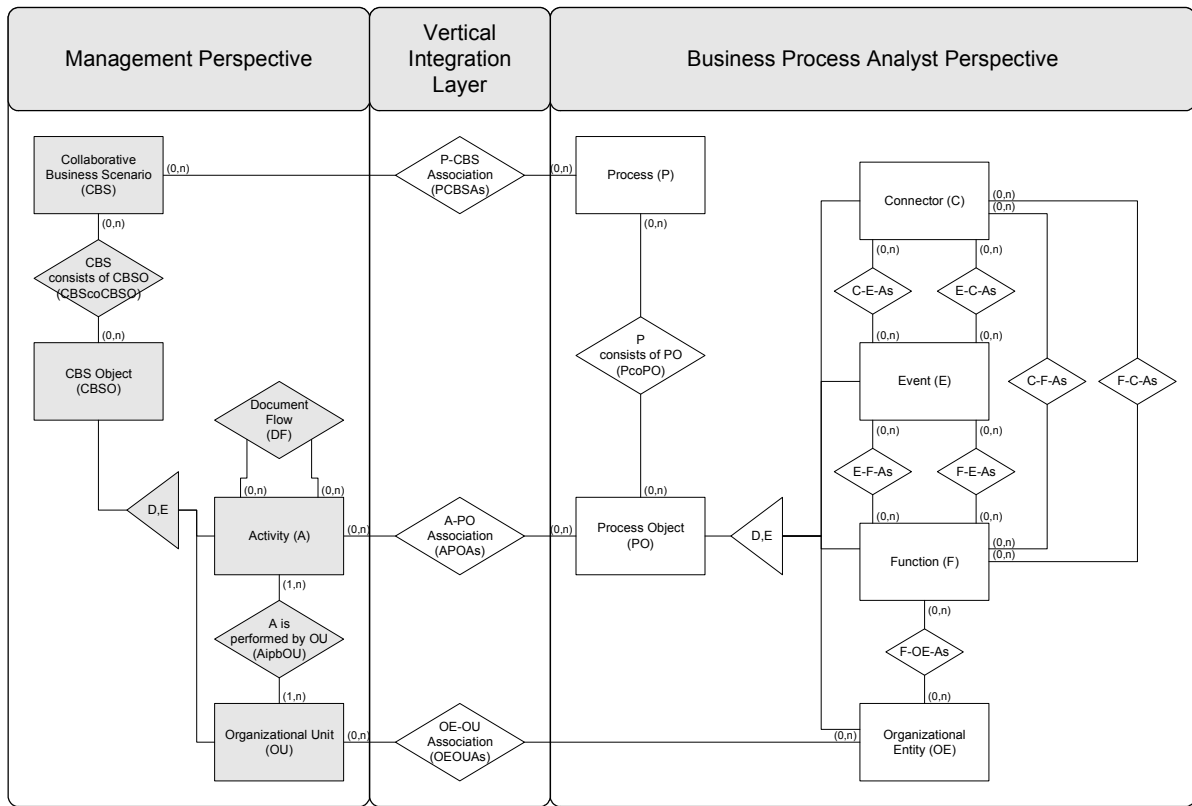


Fig. 6: Segment of the management and business process analyst perspectives of the vertically integrated meta

Given the introduced vertically integrated meta model, we can identify EPCs within the Business Process Analyst perspective affected by the configuration within the management perspective (Q4). In order to perform Q4 we need two additional relations (R8, R9): P (*Process*) (entity type, part of the EPC meta model) and $PCBSAs$ (*P-CBS Association*) (n:m relationship type, part of the integration layer between EPCs and CBS):

R8: $P = (pID, pName, pVersion)$

R9: $PCBSAs = (\underline{pcbsasID}, cbsID, pID, pcbsasName, pcbsasVersion)$

Q4: Retrieve available information about every EPC involved in the Business Process Analyst perspective for “Vendor Managed Inventory”:

$\pi_{pID, pName, pVersion} (\sigma_{cbsName="Vendor Managed Inventory"} (CBS) \triangleright \triangleleft PCBSAs \triangleright \triangleleft P)$

The derived set of elements represents the set of EPCs which are affected by the configuration. In order to enable consistency between the configured CBS and the affected, not yet configured EPCs, we furthermore need to enquire about the affected set of EPC objects within these EPCs (Q5). Again, we need two new relations (R10, R11) in order to perform Q5: *PO (Process Object)* (entity type, part of the EPC meta model) and *APOAs (A-PO Association)* (n:m relationship type, part of the integration layer between EPCs and CBS):

R10: $PO = (\underline{poID}, poName, poVersion)$

R11: $APOAs = (\underline{apoasID}, aID, poID, apoasName, apoasVersion)$

Q5: Retrieve available information about every EPC object involved in the Business Process Analyst perspective for the CBS activity “Generate Demand Forecast”:

$$\pi_{poID, poName, poVersion} (\sigma_{cbsName="Vendor Managed Inventory"}(CBS) \triangleright \triangleleft \sigma_{aName="Generate Demand Forecast"}(A) \triangleright \triangleleft CBSO \triangleright \triangleleft CBScoCBSO \triangleright \triangleleft APOAs \triangleright \triangleleft PO)$$

Since functions are connected to organizational units in EPCs, we need the set of affected functions (according to the introduced meta model a subset of EPC objects). Assuming furthermore that a CBS organizational unit corresponds to exactly one EPC organizational entity, we can enquire about the set of EPC objects (functions connected to organizational units) which require for an update due to the configuration within the management perspective (Q6). We need new four new relations to perform Q6: *F (Function)* (entity type, part of the EPC meta model), *OE (Organizational Entity)* (entity type, part of the EPC meta model), *FOEAs (F-OE Association)* (n:m relationship type, part of the EPC meta model) and *OEOUAs (OE-OU Association)* (n:m relationship type, part of the integration layer between EPCs and CBS):

R12: $F = (\underline{fID}, poID, fName, fVersion)$

R13: $OE = (\underline{oeID}, oeName, oeVersion)$

R14: $OEOUAs = (\underline{oeouasID}, oeouasName, eoID, ouID, oeouasVersion)$

R15: $FOEAs = (\underline{foeasID}, fID, oeID, foeasName, foeasVersion)$

Q6: Retrieve the set of EPC function-organizational entity relationships that are affected by the configuration within the management perspective:

$$\pi_{foeasID, fID, oeID, foeasName, foeasVersion} (\sigma_{cbsName="Vendor Managed Inventory"} (CBS) \triangleright \triangleleft CBScoCBSO \triangleright \triangleleft \sigma_{aName="Generate Demand Forecast"} (A) \triangleright \triangleleft CBSO \triangleright \triangleleft APOAs \triangleright \triangleleft PO \triangleright \triangleleft F \triangleright \triangleleft FOEAs \triangleright \triangleleft OE)$$

U6: Since U3 updated the organisational unit within the CBS after Q3, we can use the same information to update *oeID* for the derived set of elements (Changes the association from every EPC function associated to the CBS activity “Create Demand Forecast” from the Organisational Entity “Customer / Retailer” to “Vendor”). Furthermore, update *foeasVersion* for the derived set of elements (Changes the element from a reference element into a configured element)

After U6 we made sure that the configuration decision within the management perspective had an impact on the models within the business process analyst perspective. Similar to this example, other configuration scenarios can be specified and implemented. If, for instance, a CBS Activity were to be deleted at management level, Q5 can be used to query which Functions (allegedly more than one) in an EPC correspond to this activity and these items can be marked for deletion as well.

We have described in our example so far how a business process model can be configured automatically as a result of a configuration decision at management level. Similarly, we wish to configure a model that serves as (an ideally machine understandable) technical description of a business process according to changes at business process analyst level. Fig. 7 depicts such a scenario as an extension of the already discussed example. The configuration of the initial VMI Collaborative Business Scenario and its impact on the business process analyst level has already been discussed (upper left and upper middle part of Fig. 7). In the original state, the process is handed before the CBS activity “Generate demand forecast” from the Customer / Retailer to the Vendor. At the business process analyst perspective this meant a change in organizational responsibility. However, at the technical analyst level, additional functionality must be provided for conditionally converting incoming business documents. Such conversion must be performed, if the documents created by the Customer / Retailer cannot automatically be processed by the vendor’s ES. After configuring the Collaborative Business Scenario as a result of a management decision and the corresponding configuration of the EPC at business process analyst level, an integrated YAWL diagram can also be

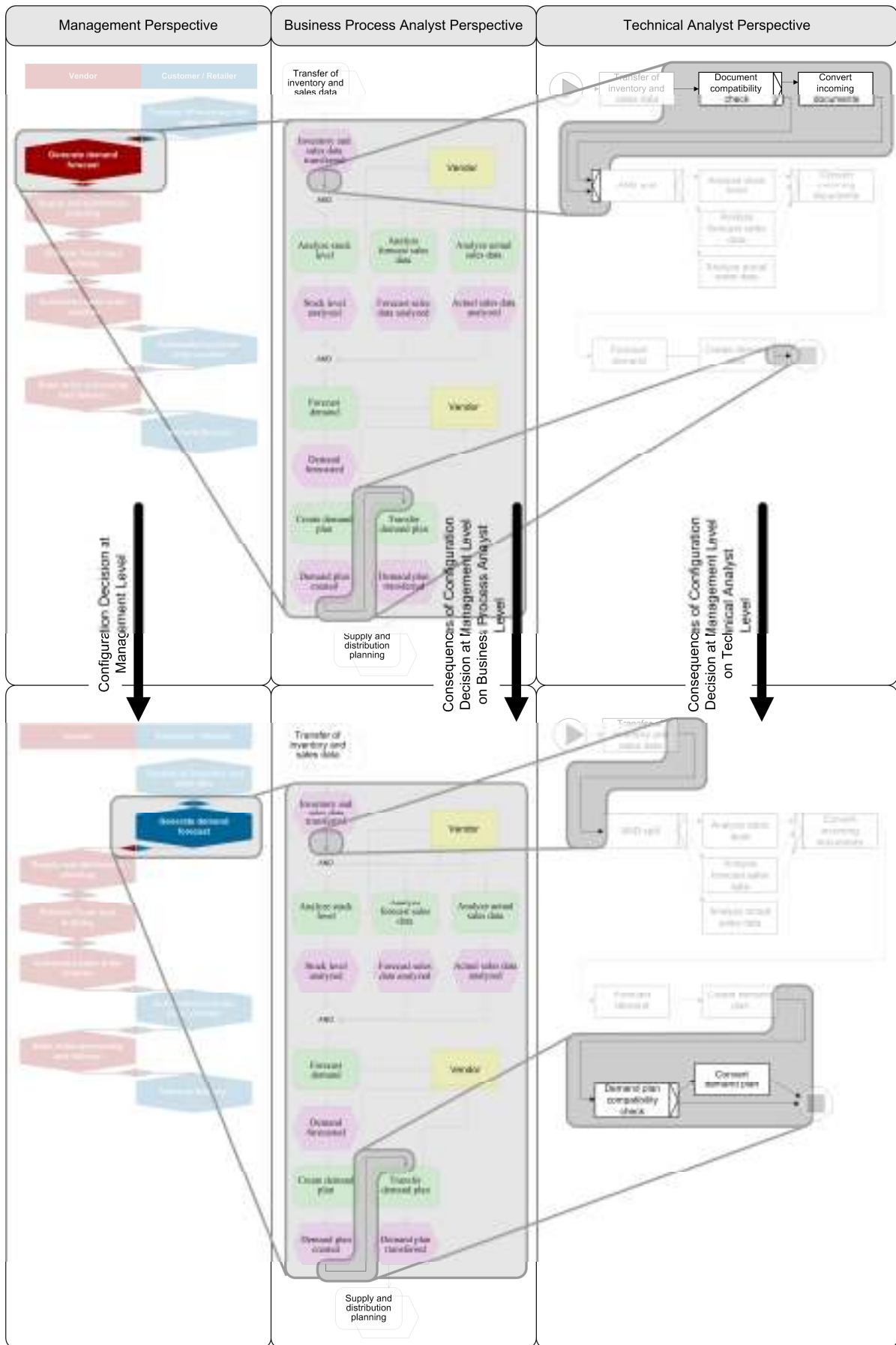


Fig. 7: The configuration decision at management, business process analyst, and technical analyst levels

adapted in order to account for the changes (lower right part of Fig. 7). After its automatic configuration the YAWL diagram does no longer include the initial document conversion step, because the process at this stage is no longer handed over before this part of the process from the Customer / Retailer to the Vendor. However, at the end of the YAWL diagram corresponding to the CBS activity “Generate demand forecast,” a conditional document conversion step must be included, because after configuration the process will be handed over from one party to another after the process has been executed according to the YAWL diagram.

In a real-world ES project a preliminary business blueprint phase is concerned with decisions at management level (such as those which lead to CBS configuration). However, before the actual ES can be configured for using it to assist daily operations, a business analysis phase must be undertaken in order to determine the exact requirements that the ES must meet after it will ‘go live.’ As part of this phase the business process models describing ES capabilities will be configured if the generic description of the model does not fit the specific organizational requirements. With respect to the topic of this paper two major developments can be distinguished in this phase:

1. The business process models that were initially used to describe ES capabilities from a business process analyst perspective have been changed in patterns that were predefined by the ES vendor. Such configuration takes place if a vendor builds in choices that can be made at configuration time (e.g., which retail type must be supported, what type of warehouses are necessary, etc.). In this case an integration layer between the business process analyst and technical analyst perspectives that is built similarly to the one that we discussed above between the management and the business process analyst perspectives can take care of an automatic configuration of models at the technical analyst level as shown in Fig. 7.
2. The business process models that were initially used to describe ES capabilities from a business process analyst perspective have been changed to an extent that the integration layer between the business process analyst and technical analyst perspectives is not able to handle automatic configuration of models at technical analyst level anymore. Given the complexity of models at the business process analyst and technical analyst perspectives and, generically

speaking, the variety of means of describing business processes from these perspectives, this is not unlikely to occur. In these cases we have included support for re-generating models from the technical analyst perspective in order to enable consistency between the layers.

Because the second case is realistic given the individuality of specific organizations and the generality of ES, it requires more attention. EPCs are useful for the purpose of communicating a process between business process analysts, but they are unsuitable as a description of a process from the technical analysis perspective. More specifically, EPCs cannot be used at the technical analyst level anymore as a result of their informal semantics [27]. It has been pointed out that many constructs are ambiguous [2, 16, 28]. In [28] an attempt is made to provide a suitable semantics to the so-called XOR and OR joins. Although this is possible, in real life applications humans tend to interpret EPCs in a less precise manner. Therefore, approaches such as the one in [16] have been proposed to migrate an informal EPC into an executable model.

However, before transforming a model at the business process analyst level (e.g., an EPC) to the technical analyst level, it is important to assess the correctness of the EPC. The so-called ProM framework is an open source tool that provides support for this task (www.processmining.org) and other tasks such as discovering processes based on event logs (e.g., the transaction logs or audit trails of an information system). The discovered model can be represented in a variety of languages (including EPCs) and consequently be exported to YAWL. ProM also allows for conformance checking, i.e., comparing a process model or process configuration with the actual behavior as registered by the information system. Clearly, these functionalities are very useful for the suggested transition of models at the business process analyst level to models at the technical analyst level in cases where configuration at the business process analyst level exceeds predefined configuration patterns. Fig. 8 shows a screenshot of ProM while analyzing the initial EPC (before configuration). As the figure shows, the EPC contains no errors. ProM uses an algorithm that is able to deal with multiple interpretations of an EPC [18]. However, in this case the model is unambiguous. Therefore, no further refinements are necessary.

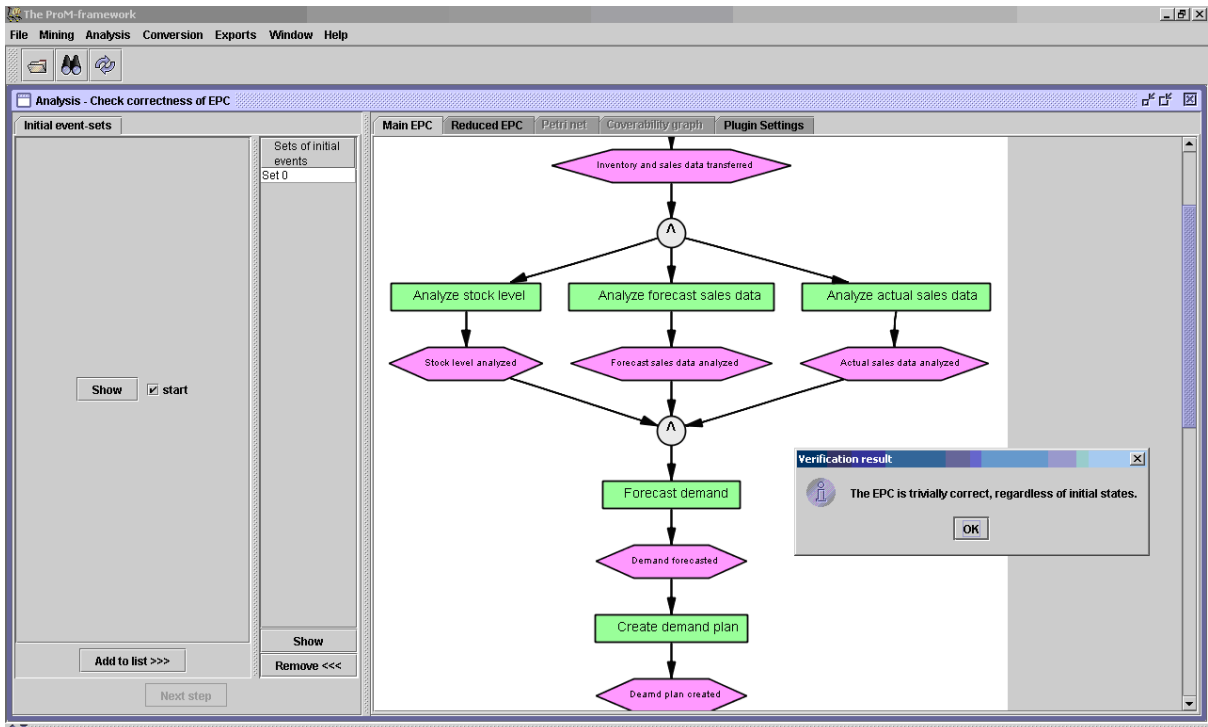


Fig. 8: Screenshot of ProM while analyzing the EPC shown in Fig. 7

ProM is also able to transform a correct EPC into a YAWL specification as shown in Fig. 9. Clearly, the mapping of EPCs onto YAWL does not add any information, i.e., technically related information needs to be added later. ProM only generates an initial model. However, this model can be executed immediately by the YAWL engine.

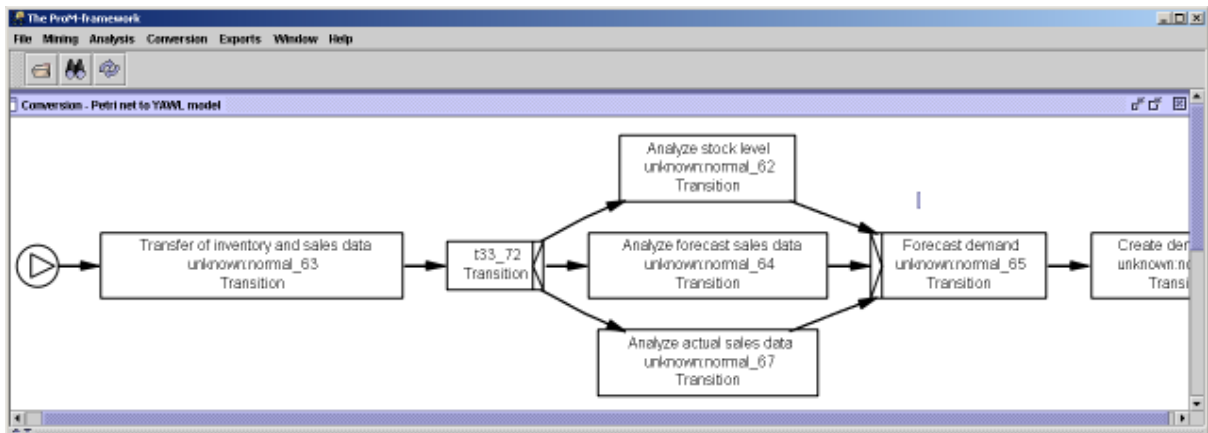


Fig. 9: Screenshot of ProM while transforming the EPC into a YAWL model

Fig. 10 shows a screenshot of the YAWL work-list handler while executing three process instances (cases) according to the configured process model. Each of the three cases is in the state where the inventory and sales data have been transferred and the three subsequent steps are enabled (i.e., 9 work items are waiting to be processed).

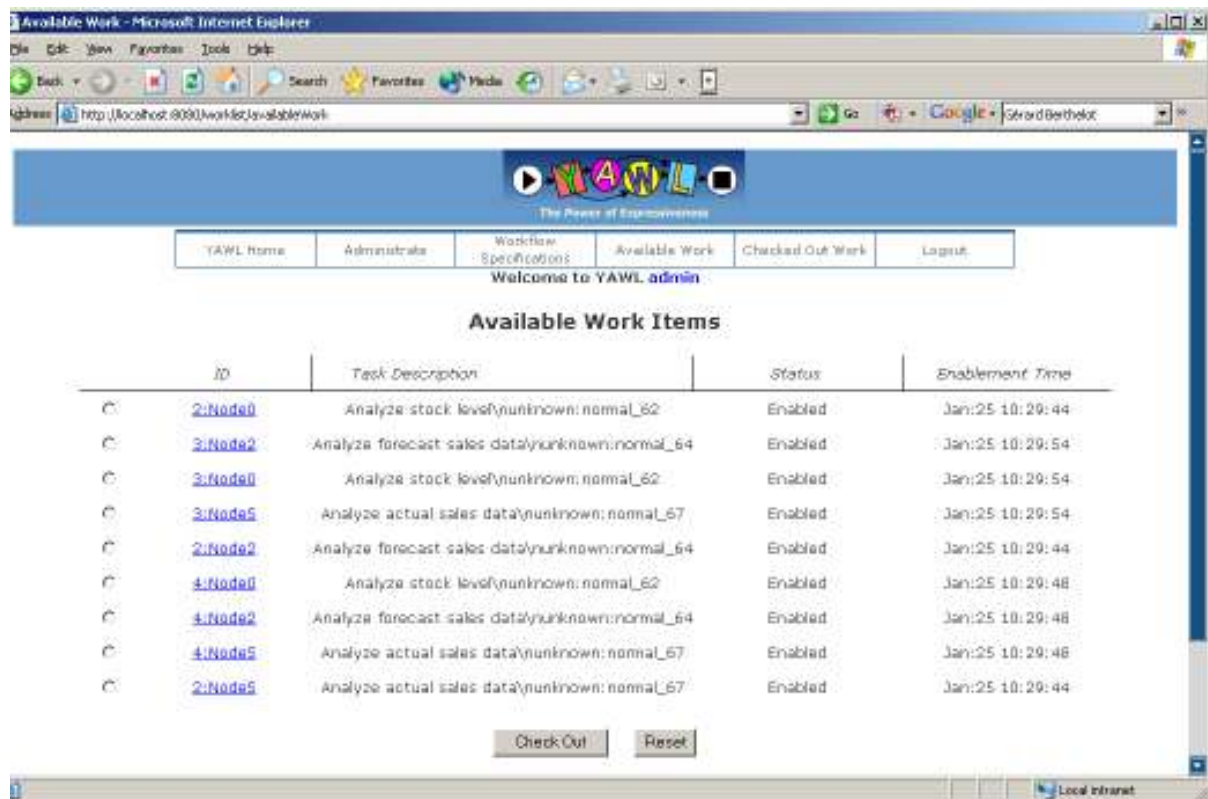


Fig. 10: Screenshot of the YAWL work-list handler while executing the model depicted in Fig 7 (YAWL model in lower right part) after the configuration decision

In this section, we discussed a concrete example involving three modeling techniques each of which depicts a single business process from three different perspectives: management, business process analyst, and technical analyst. We suggested integrating these three languages by means of integrating their language constructs at meta level. We discussed how we can create impact of configuration decisions at one level on models at another level. We introduced relational algebra expressions that can be used to retrieve relevant constructs of models impacted by configuration decisions at another level. We also discussed that in cases where configuration escapes the suggested integration layer by means of meta data links other mechanisms are necessary such as the transformation of EPC models

to a concrete workflow management system (YAWL) and showed that there is concrete tool support. ProM is able to generate a YAWL model based on an EPC and this model can immediately be executed in the YAWL environment. Note that in reality, however, more technical aspects need to be added (e.g., location of data, application integration, transactional properties, etc.).

6. Limitations of the Approach

There are several factors that limit our approach and therefore need to be discussed. First and foremost, our propositions necessarily increase the modeling effort that typically is already large in ES projects. We do not target this issue directly in this paper but work on contributions to overcome this problem in related projects. It must be clear that our approach is only intended to be applicable in scenarios where conceptual models are bound to process execution of an ES. Only in these cases an increased modeling effort is justifiable. Furthermore, we investigate the problem of “modeling in the large” in a related project (funded by the Australian Government and SAP Research) in order to better understand the problems that result from large modeling projects and to deliver value propositions for managing such projects.

Another problem we need to address in further research is related to the integration layers between the management, business analyst, and technical analyst layers. The problem results from using established modeling languages and integrating them instead of starting from the scratch with a one-language or integrated-language environment for ES projects. We typically map a language construct of one language with a similar language construct of another language at a different level with a many-to-many relationship. Again, this becomes necessary because the involved languages have been developed with different purposes in mind and for different user groups. They therefore depict entirely different parts of identical processes with entirely different semantics. Therefore, deleting an item in one model does not lead to an automatic deletion of corresponding items in a model on another level. Nevertheless, the approach helps in identifying items and marking them for deletion subject to discussion on another level. If deletion is impossible, then a feedback mechanism is gained which informs respective actors.

A third limitation addresses the used modeling languages. We picked common and well-established languages for integrating them. Other languages on the introduced levels might be suited as well or even better suited for the task that we performed. We do not claim to have found the best possible integration scenario. We rather argued that such vertical integration is necessary and showed conceptually that it is possible and how it can be achieved.

7. Conclusions and Outlook

Configuration is one of the most resource-consuming ES implementation phases, with considerable space for improvement. Our approach targets an increased efficiency of ES configuration by vertically integrating existing process modeling languages that have evolved for providing process information to different user groups. We argued that vertically integrated models need to be configurable and introduced an approach for such configurations. Both integration and configurability become necessary because configuration can be undertaken at management, business process analyst, and technical analyst levels and configuration should not be undertaken redundantly. We introduced a business example which outlines our approach. Together with the vision that comprehensive ES software fully acts according to specified process models, which is, for instance pursued with SAP's NetWeaver, our approach allows for efficiently configuring such software.

Our further work will mainly consist of two directions. First, we will work on a prototype which enables model configuration in the way we proposed it. Second, we will conduct empirical studies, for understanding which languages need to be integrated for configuring which ES packages most efficiently.

References

- [1] W.M.P.van der Aalst, Formalization and Verification of Event-driven Process Chains, Information & Software Technology 41(10) (1999).

- [2] W.M.P. van der Aalst, J. Desel, and E. Kindler, On the Semantics of EPCs: A Vicious Circle, in: M. Nüttgens and F.J. Rump, Eds., Proceedings of the EPK 2002: Business Process Management using EPCs (Gesellschaft für Informatik, Trier, Germany, 2002).
- [3] W.M.P. van der Aalst and A.H.M. ter Hofstede, YAWL: Yet Another Workflow Language, Information Systems, 30(4) (2005).
- [4] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, Workflow Patterns, Distributed and Parallel Databases 14(1) (2003).
- [5] N. Bancroft, H. Seip, and A. Sprengel, Implementing SAP R/3: How to introduce a Large System into a Large Organisation (2nd ed, Manning Inc, Greenwich, 1998).
- [6] A. Basu and A. Kumar, Research Commentary: Workflow Management Issues in e-Business, Information Systems Research 13(1) (2002).
- [7] C. Bechtel and J. Jayaram, Supply Chain Management: A Strategic Perspective, The International Journal of Logistics Management 8(1) (1997).
- [8] J. Becker, P. Delfmann, A. Dreiling, R. Knackstedt, and D. Kuropka, Configurative Process Modeling – Outlining an Approach to increased Business Process Model Usability, in: 2004 IRMA International Conference (Idea Group Publishing, New Orleans, LA, 2004).
- [9] J. Becker, P. Delfmann, and R. Knackstedt, Konstruktion von Referenzmodellierungssprachen. Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle, Wirtschaftsinformatik 46(4) (2004).
- [10] J. Becker and R. Schütte, Handelsinformationssysteme, (2nd ed, Moderne Industrie, Landsberg (Lech), Germany, 2004).
- [11] L. Brehm, A. Heinzl, and M.L. Markus, Tailoring ERP Systems: A Spectrum of Choices and their Implications, in: 34th Hawaii International Conference on System Sciences (HICSS) (Hawaii, 2001).
- [12] P.P. Chen, The Entity-Relationship Model: Towards a Unified View of Data, ACM Transactions on Database Systems 1(1) (1976).
- [13] M. Christopher, Logistics and Supply Chain Management. Strategies for Reducing Cost and Improving Service, (2 ed, Financial Times Professional, London, 1998).

- [14] E.F. Codd, A Relational Model of Data for Large Shared Data Banks, *Communications of ACM* 13(6) (1970).
- [15] T.H. Davenport, Putting the Enterprise into the Enterprise System, *Harvard Business Review* 76(4) (1998).
- [16] J. Dehnert and W.M.P. van der Aalst, Bridging the Gap Between Business Models and Workflow Specifications, *International Journal of Cooperative Information Systems*, 13(3) (2004).
- [17] S.M. Disney and D.R. Towill, The Effect of Vendor Managed Inventory (VMI) Dynamics on the Bullwhip Effect in Supply Chains, *International Journal of Production Economics* 85(2) (2003).
- [18] B.F. van Dongen, H.M.W. Verbeek, and W.M.P. van der Aalst, Verification of EPCs: Using Reduction Rules and Petri Nets, in: *Proceedings of the Conference on Advanced Information Systems Engineering (CAiSE 2005)*, volume 3520 of *Lecture Notes in Computer Science* (Springer, Berlin, 2005).
- [19] M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software through Process Technology* (Wiley & Sons, New York, 2005).
- [20] C.F. Gibson, C.J. Singer, A.A. Schnidman, and T.H. Davenport, Strategies for Making an Information System Fit Your Organization, *Management Review*, 73(1) (1984).
- [21] P. Green and M. Rosemann, Integrated Process Modeling: An Ontological Evaluation, *Information Systems* 25(3) (2000).
- [22] F. Hewitt, Supply Chain Redesign, *The International Journal of Logistics Management* 5(2) (1994).
- [23] C.P. Holland and B. Light, A Critical Success Factors Model For ERP Implementation, *IEEE Software* 16(3) (1999).
- [24] R. Holten, *Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz* (Gabler, Wiesbaden, Germany, 1999).

- [25] R. Holten, Specification of Management Views in Information Warehouse Projects, *Information Systems* 28(7) (2003).
- [26] R. Holten, A. Dreiling, and J. Becker, Ontology-Driven Method Engineering for Information Systems Development, in: P. Green and M. Rosemann, Eds., *Business Systems Analysis with Ontologies* (IDEA Group Publishing, Hershey, PA, 2005).
- [27] G. Keller, M. Nüttgens, and A.W. Scheer. Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). *Veröffentlichungen des Instituts für Wirtschaftsinformatik, Vol. 89 (in German)* (University of Saarland, Saarbrücken, 1992).
- [28] E. Kindler, On the Semantics of EPCs: A Framework for Resolving the Vicious Circle, *Data and Knowledge Engineering*, 56(1) (2006).
- [29] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained* (Addison-Wesley, Boston, 2003).
- [30] O.I. Lindland, G. Sindre, and A. Sølvsberg, Understanding Quality in Conceptual Modelling, *IEEE Software* 11(2) (1994).
- [31] H.C. Lucas Jr., L.N. Stern, E.J. Walton, and M.J. Ginzberg, Implementing Packaged Software, *MIS Quarterly* 12(4) (1988).
- [32] R. Luxem, The Impact of Trading Digital Products on Retail Information Systems, in: 33rd Hawaii International Conference on System Sciences (Maui, HI, 2000).
- [33] Merriam-Webster, Merriam-Webster's Collegiate Dictionary (online version available at <http://www.m-w.com>), (11th ed, Merriam-Webster, Springfield, MA, USA, 2003).
- [34] RosettaNet, <http://www.rosettanel.org> (2004).
- [35] SAP AG, Collaborative Business Scenario "Vendor Managed Inventory" (SAP AG, 2004).
- [36] A.-W. Scheer, *ARIS – Business Process Modeling*, (3 ed, Springer, Berlin, 2000).
- [37] A.-W. Scheer, *Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse*, (7 ed, Springer, Berlin, 1997).
- [38] P. Soffer, B. Golany, and D. Dori, ERP Modeling: A Comprehensive Approach, *Information Systems* 28(6) (2003).
- [39] T. Stein, SAP sued over R/3, *InformationWeek* (698) (1998).

- [40] S. Strahringer, Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden (Shaker, Herzogenrath, Germany, 1996).
- [41] M. Sumner and J. Hamilton, The Turnaround ERP project: Strategies and Issues, in: 11th Annual Americas Conference on Information Systems (AMCIS) (Omaha, NE, 2005).
- [42] Supply Chain Council, Supply-Chain Operations Reference-Model. Overview of SCOR Version 5.0 (Supply Chain Council, Pittsburgh, PA, 2001).
- [43] H.M. Tan, The Actual Practice of Modelling in SAP Projects (Honors Thesis, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia 2004).
- [44] TeleManagement Forum, NGOSS Technical Overview, TeleManagement Forum, Presentation TMFC1852 (2004).
- [45] Y. Wand and R. Weber, On the Deep Structure of Information Systems, Information Systems Journal 5(3) (1995).
- [46] R. Weber, Ontological Foundations of Information Systems (Coopers and Lybrand, Melbourne, Australia, 1997).
- [47] R. Whitley, On the Nature of Managerial Tasks and Skills: Their distinguishing Characteristics and Organization, Journal of Management Studies 26(3) (1989).
- [48] Workflow Management Coalition (WFMC), www.wfmc.org (2004).

About the Authors



Dr. Alexander Dreiling is a Researcher at SAP Research (CEC Brisbane, Australia). He received his MSc and PhD in Information Systems from the University of Münster in Germany. Prior to joining SAP Research he worked (research-related) for the European Research Center for Information Systems in Münster, Germany and Queensland University of Technology in Brisbane, Australia. Alex is primarily interested in conceptual modeling, in particular business process modeling, data warehousing, and business-related information systems research. His research so far led to more than 25 journal papers, conference papers, and book chapters.



Dr. Michael Rosemann is a Professor for Information Systems and Co-Leader of the Business Process Management Group at Queensland University of Technology, Brisbane. His main areas of research are business process management, business process modeling, Enterprise Systems and ontologies. Besides more than 120 refereed papers including publications in journals such as Information Systems, European Journal of Information Systems and IEEE Transactions on Knowledge and Data Engineering, he is the author and editor of five books. Michael is a member of the Editorial Board of six journals and a member of the ARC (Australian Research Council) College of Experts.



Dr. Wil van der Aalst is a Professor of Information Systems and head of the Information Systems department of the Technische Universiteit Eindhoven. He is also an Adjunct Professor at the Faculty of Information Technology of Queensland University of Technology. His research interests include Business Process Management, information systems, simulation, Petri nets, process models, workflow management systems, process mining, verification techniques, enterprise resource planning systems, computer supported cooperative work, and inter-organizational business processes. He published more than 200 books, journal papers, book chapters, conference papers, and reports on these topics.