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

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

Michael Rosemann  
Centre for IT Innovation,  
Queensland University of Technology  
126 Margaret St.  
Brisbane Qld 4000, Australia  
m.rosemann@qut.edu.au

Wil M. P. van der Aalst  
Department of Technology Management, Eindhoven University of Technology  
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands  
w.m.p.v.daalst@tm.tue.nl

Abstract
The configuration of comprehensive Enterprise Systems to meet the specific requirements of an organisation up to today is consuming significant resources. The results of failing implementation projects are severe and may even threaten the organisation’s existence. This paper proposes a method which aims at increasing the efficiency of Enterprise Systems implementations. First, we argue that existing process modelling 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 motivate that an integrated process model based on the integrated meta model needs to be configurable and elaborate on the mechanisms by which this model configuration can be achieved. We introduce a business example using SAP modelling techniques to illustrate the proposed method.

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

1. Introduction
The common presupposition of Enterprise Systems (ES) is that they support organisations in their operations and lead to significant efficiency gains. This is only true for well-implemented ES that support an organisation’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 (Stein, 1998). Other examples include Mobil Europe and Dow Chemical both of which spent hundreds of millions of US$ for ES implementation (Davenport, 1998).

As long as ES vendors will continue to develop generic, off-the-shelf ES packages, this problem will prevail because organisations have a non-generic or individual character. Within academia this development is reflected by a constantly growing body of literature on configuration (examples include Bancroft et al., 1998; Brehm et al., 2001; Davenport et al., 1998; Gibson et al., 1984; Holland and Light, 1999; Lucas Jr. et al., 1988; Soffer et al., 2003) emphasising that information systems are typically not implemented in an organisational context but adapted to organisational needs from ‘off-the-shelf’ packages.

This paper introduces a method which targets increased usability of conceptual modelling for the purpose of ES configuration as conceptual modelling is underutilised in this context (Tan, 2004). One of the reasons for this is that modelling 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 modelling is used for requirements engineering purposes, usually the models do not automatically impact on the software configuration which again drives the perception that modelling is an overhead. Since modelling is underutilised the question arises as to how to create an improved value proposition related to conceptual modelling as part of an ES project. This paper’s approach to achieve this goal features three different aspects:

1. Various perspectives of modelling: Managers and technical project members have a different perspective on a business process. To meet the requirements of different user groups, alternative modelling languages have evolved. Changing established modelling languages is time-consuming and may result in resistance of project members to use modelling. We therefore propose to integrate existing process modelling languages.

2. Model configuration: A set of predefined conceptual models needs to be adapted to the specific requirements of an organisation.

3. ES configuration by means of model configuration: Usually, ES software needs to be adapted to the specific requirements of an organisation.

The model integration we propose differs from integration concepts underlying techniques such as UML or ARIS. We propose to integrate process modelling 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 business example that will provide a better understanding of our approach in section 4. Finally, a short outlook will be given and future prospects will be discussed.

2. Perspectives in Process Modelling

Within the fields of Information Systems and Computer Science numerous process modelling languages have evolved. These techniques vary in their degree of comprehensibility to certain user groups, i.e., they are of different pragmatic quality (Lindland et al., 1994). Some process modelling 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 modelling experiences is more important than the use of a meta model with high expressive power. Other modelling techniques describe a business process with the purpose of executing the process automatically (workflow languages). Such techniques demand a high rigor in terms of the meta model, but are often only used by a limited number of experienced modellers. We distinguish between the three perspectives management, business process analyst, and technical analyst and discuss them in more detail. This distinction follows the commonly accepted distinction between managerial and non-managerial work on the one hand (Whitley, 1989), and between business and IT on the other hand.

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 organisation including related inter-organisational business processes. Since the management is responsible for the entire set of business processes, a certain level of abstraction is required in order to compensate the complexity which arises from the significant number of models.
Business process frameworks as the highest level of an enterprise-wide model provide a glance at the entire set of business processes within an organisation. 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 (Scheer, 1997). The Retail-H, as another example, depicts the business processes involved in retailing (Becker and Schütte, 2004; Luxem, 2000). 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 (TeleManagement Forum, 2004).

Several ES software providers have included modelling 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 modelling. 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 modelling 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 modelling languages have been developed to address the needs of this perspective. For instance, Event-driven Process Chains (EPCs) as an integrating modelling language of the process perspective within the ARIS approach (Scheer, 2000) 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, and many software products are available, there is no commonly accepted standard of a workflow language. Modelling standards such as BPEL4WS (Business Process Execution Language for Web Services) or BPML (Business Process Modelling Language) or notations such as BPMN (Business Process Modelling 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 (Basu and Kumar, 2002) or on workflow languages in particular (van der Aalst and ter Hofstede, 2005) based on a rigorous analysis of workflow language
requirements (van der Aalst et al., 2003) suggests that this domain will change significantly over the next years.

Several organisations proposed workflow standards such as the Workflow Management Coalition (WFMC) (Workflow Management Coalition (WFMC), 2004), the RosettaNet Consortium (RosettaNet, 2004), or the Supply Chain Council (Supply Chain Council, 2001) towards workflow architectures, languages, or specific process schemas.

We selected YAWL (van der Aalst and ter Hofstede, 2005) for our research. YAWL is a workflow language based on a thorough analysis of workflow patterns (van der Aalst et al., 2003). Especially at an instance level, YAWL features several sophisticated modelling constructs to handle splits and synchronisation of workflow branches.

3. Integration of Perspectives
The existence of different process modelling languages in practice confronts an organisation with either accepting and building upon their existence or converting all users to a single language. We argue that the latter is not an option for two main reasons: it is impossible if a language is incapable of expressing constructs that some users want to express. It is furthermore cumbersome or impracticable to achieve a single-language environment if different languages are established within different user groups and some users must convert, because not everyone will appreciate changes. Hence, we propose to tolerate the co-existence of different languages and to integrate languages that express business processes at different granularity levels.

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 the language constructs within the perspectives as shown in Fig. 1.

![Diagram](image)

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

One of the major advantages gained by an integration of languages of the three introduced perspectives is the impact of configuration decisions between the levels. If a top-down approach is chosen within a 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.

Figure 2 sketches the impact of configuration decisions at higher levels on models at lower levels. In the example a single business process within an SAP Solution Map (one box in the left column) refers to an entire SAP Collaborative Business Scenario (middle column). Some activities within the CBS then refer to parts of processes within Event-driven Process Chains (right column).

Integrating languages featuring a different level of abstraction is naturally bound to losing information towards the more abstract information modelling 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. E.g. the Bunge-Wand-Weber Ontology provides a useful framework of such an ontological evaluation (Wand and Weber, 1995; Weber, 1997). Language constructs belonging to the same ontological category (e.g. thing, transformation, or state) can be mapped relatively easy at a meta level. This mapping will usually lead to one-to-many relationships between the statements made in the languages rather than one-to-one relationships. As an example the language construct Function of Event-driven Process Chains can be mapped to an activity in a Collaborative Business Scenario. In a detailed EPC model one CBS activity will usually be
represented by many EPC functions. Table 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.

<table>
<thead>
<tr>
<th></th>
<th>SM, CBS</th>
<th>EPC*</th>
<th>YAWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>x</td>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>Property</td>
<td>-</td>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>State</td>
<td>-</td>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>Transformation</td>
<td>x</td>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>Stable State</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Tab. 1. Evaluation of modelling languages with the base constructs of the Bunge-Wand-Weber Ontology (Wand and Weber, 1995). *a detailed ontological analysis of EPCs can be found in (Green and Rosemann, 2000).

The ontological evaluation of the modelling languages is especially useful as it reveals the incapability of ‘high-level’ languages to make statements about aspects that only can be formulated with more complex languages. 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 modelling language features an ontological construct which is not supported in a less detailed modelling 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 modelling languages to depict various aspects of control flow reveals that integrating the languages for different perspectives is a non-trivial task. An evaluation of the Workflow Patterns supported by the introduced languages shows that towards the more abstract languages the control flow is only insufficiently supported (comp. Tab. 2).

<table>
<thead>
<tr>
<th>Workflow Pattern</th>
<th>SM, CBS</th>
<th>EPC*</th>
<th>YAWL*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>x/-**</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Parallel Split</td>
<td>x/-**</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Synchronisation</td>
<td>x/-**</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Exclusive Choice</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Simple Merge</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multi-Choice</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Synchronising Merge</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multi-Merge</td>
<td>-</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>Discriminator</td>
<td>-</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>Arbitrary Cycles</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
**Workflow Pattern** | SM, CBS | EPC* | YAWL*
--- | --- | --- | ---
Implicit Termination | x/-** | x | -
Multiple Instances Without Synchronisation | - | - | 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
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 (van der Aalst and ter Hofstede, 2005).**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.

In order to vertically integrate SMs, CBSs, EPCs, and YAWL their language constructs need to be matched at a meta level, since the meta models of the included languages are comprised of these language constructs and relationships between them. For the remainder of this paper we will show how vertical integration of process modelling languages can be achieved using a business example after making some brief remarks on process configuration.

4. Business Example
4.1 Preliminary Remarks on Process Configuration

Configuration and customisation are often used interchangeably. Merriam-Webster's Collegiate Dictionary (2003) defines configuration as the “relative arrangement of parts or elements” whereas customising 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 customisation (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 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 organisation with generic enterprise systems in order to meet the business requirements of the enterprise in the most efficient way. For the sake of simplicity use the term configuration instead of reconfiguration in this paper.

All of our configuration mechanisms are anchored at meta level. This means that we achieve configuration by manipulating a meta-data repository (for instance a relational database) which contains data about process models. In other words: we are not deleting, e.g., a function within a process model but an entry within the ‘function’ table of a meta-data repository. This configuration affects the type level (i.e. the model will be configured)
because the information stored within the meta-data repository is 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 meta-data repository)
- **Delete**: an object is removed from the reference model during configuration (an entity is removed from the meta-data repository)
- **Add**: an object is added to the reference model during configuration (an entity is added to the meta-data 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
- **Unify**: deletes a number of objects to be unified and adds one unified object to the model
- **Change**: deletes an object and adds another one to the model which represents an alternative to the original object

### 4.2 Business Scenario

In order to illustrate the proposed configuration approach we will now provide a short business example from the domain of Supply Chain Management (SCM), which is concerned with the design, operation, and maintenance of integrated value chains. The main objectives of SCM are the satisfaction of customer needs while simultaneously maximising customer service (Bechtel and Jayaram, 1997, Christopher, 1998, Hewitt, 1994). 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 (Disney and Towill, 2003). For our discussion we will use an SAP example outlined in Fig. 3.
In our example a company facing volatile demands wishes to have Vendor Managed Inventory. However, the Demand Forecast should not be done by the vendor as described by SAP’s reference model (SAP AG, 2004) but in-house, because it is perceived as a competitive advantage by the company. The scenario requires for changing SAP’s reference 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 specialised disjoint and equivocally in Activity (A) and Organisational Unit (OU). Activities are performed by organisational units. Finally, 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.

![Management Perspective](image)

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. The following seven relations (R1-R7) can be defined according to the meta model segment from Fig. 4 (we have underlines key attributes and abstracted from attributes which would be necessary in a real setting such as time frame, cost, etc.).

R1:  \[CBS = (cbsID, cbsName, cbsVersion)\]
R2:  \[CBSO = (cbsoID, cbsoName, objecttype, cbsoVersion)\]
R3:  \[CBScoCBSO = (cbscochsoID, cbscochsoName, cbscochsoVersion)\]
R4:  \[A = (aID, cbsoID, aName, aVersion)\]
R5:  \[OU = (ouID, cbsID, ouName, ouVersion)\]
R6:  \[AipbOU = (aipbouID, aID, ouID, aipbouName, aipbouVersion)\]
R7:  \[DF = (dfID, previousAID, subsequentAID, dfName, document, dfVersion)\]

The configuration 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:
\[
\pi_{\text{aID}, \text{cbsName}, \text{aVersion}} (\sigma_{\text{aName} = \"Generate Demand Forecast\"}(A) \bowtie CBSO \bowtie CBScoCBSO \bowtie \\
\sigma_{\text{cbsName} = \"Vendor Managed Inventory\"}(CBS))
\]

**U1:** Update a *Version* 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:
\[
\pi_{\text{dfID, previousAID, subsequentAID, dfName, document, dfVersion}} (\sigma_{\text{aName} = \"Generate Demand Forecast\"}(A) \bowtie DF \bowtie \\
CBSO \bowtie CBScoCBSO \bowtie CBS \bowtie \\
\sigma_{\text{cbsName} = \"Vendor Managed Inventory\"}(CBS))
\]

**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 - organisational unit association (AipbOU):
\[
\pi_{\text{aID}, \text{cbsID}, \text{aVersion}} (\sigma_{\text{aName} = \"Generate Demand Forecast\"}(A) \bowtie CBSO \bowtie CBScoCBSO \bowtie \\
\sigma_{\text{cbsName} = \"Vendor Managed Inventory\"}(CBS))
\]

**U3:** Update ouID for the derived set of elements (Changes the association from the activity “Create Demand Forecast” from the Organisational 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.

The configuration so far affected the management perspective solely. Without vertical integration as proposed in the last section the configuration steps would have to be performed again within the business process analyst perspective which quickly leads to a large overhead of modelling especially within large-scale requirements engineering projects. Fig. 5 thus depicts a segment of the populated framework from Fig. 1 integrating the management and business process analyst perspectives.
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):

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

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

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

\[
\pi_{pID,pName,pVersion}(\sigma_{cbsName="Vendor Managed Inventory"}^{CBS}) \bowtie \bowtie PCBSAs \bowtie \bowtie P
\]

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) with which we can perform Q5:

R10: \( PO = (poID, poName, poVersion) \)

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

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

\[
\pi_{poID,poName,poVersion}(\sigma_{cbsName="Generate Demand Forecast"}^{CBS}) \bowtie \bowtie \sigma_{cName="Generate Demand Forecast"}^{A}) \bowtie \bowtie CBSO \bowtie \bowtie CBScoCBSO \bowtie \bowtie APOAs \bowtie \bowtie PO
\]

Since functions are connected to organisational 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 organisational unit corresponds to exactly one EPC organisational entity, we can enquire about the set of EPC objects (functions connected to organisational
units) which require for an update due to the configuration within the management perspective (Q6). Again we need new relations to perform Q6:

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

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

R14: \( OEUAs = (\text{oeouasID}, \text{oeouasName}, \text{oeID}, \text{ouID}, \text{oeouasVersion}) \)

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

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

\[
\pi_{\text{foeasID}, \text{fID}, \text{oeID}, \text{foeasName}, \text{foeasVersion}} \left( \sigma_{\text{chtsName} = "Vendor Managed Inventory"} (\text{CBS}) \bowtie \text{CBScoCBSO} \right) \bowtie \ldots
\]

U6: Since U3 updated the organisational unit within the CBS after Q3, we can use the same information to update \( \text{oeID} \) 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 \( \text{foeasVersion} \) 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.

5. 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 modelling 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


