Abstract

Customization of Enterprise Systems (ES) is often a time and cost consuming task. Therefore many of these ES, such as SAP R/3, are provided with preconfigured processes and data objects that can be regarded as best-practice and which are documented by reference information models. Customization can be accelerated if these models are used for the customization process. This requires profound configuration and adaptation mechanisms that encompass variants due to specific customer needs. To ensure continuous improvement towards a shorter customization time and reduced cost, controlling of the adaptation process becomes a crucial task. This controlling aims at improving the reference model basis and their configuration mechanisms. Therefore we introduce configuration mechanisms that can be used to adapt reference information models due to company characteristics. Afterwards the configuration mechanisms are interpreted by the ES to perform the necessary customization steps.

1 Introduction, Related Work

1.1 Introduction

The introduction of comprehensive Enterprise Systems (ES) is often an expensive and time consuming task (cf. also in the following [1]). In order to serve a preferably large market, ES vendors offer a lot of functionalities so as to cope with a large amount of business requirements. The challenge is to align the business requirements of the regarded company with the provided functionality of the ES [1, 2].

As a consequence, two major tasks have to be fulfilled at an ES introduction: First, business processes of the enterprise have to be reorganized in order to streamline them and make them fit to the ES. This affects not only processes to be implemented as ES functionality but also surrounding processes that are performed manually or which serve as interfaces between employees and the ES respectively. Secondly, the ES has to be adapted to special requirements of the enterprise. Both tasks are related to extensive costs. For instance the SAP system has over 3,000 configuration tables for customizing the ES [1]. For this, specialized knowledge is needed which often results in severe consulting costs. Similarly, the reorganization of the surrounding business processes of the regarded company causes expenses spent on reorganization knowledge. Hence, in order to decrease these costs, a strategy is needed that allows an effective and efficient customizing of ES as well as an effective and efficient reorganization of surrounding business processes without the need of the mentioned in-depth knowledge.

Detailed knowledge of ES customizing and business process reengineering can be stored in information models, since they are recognized as knowledge repositories [3]. In order to reduce ES introduction cost, special information models are needed that contain existent “best practice” or “common practice” knowledge [2]. These special information models are called reference models and can provide a basis for ERP customization [2]. Nowadays, reference models are used for documentation purposes of ES and as a basis for business process change.

ES introduction can only be supported efficiently by reference models if the models can be easily adapted to specific application contexts [2] – first in order to guarantee a proper fit-of-use of the reference model according to the organizational needs of the company, and secondly in order to be able to perform the customizing process of the ES easily. Manual adaptation of the models, and in turn, of the ES, could lead to economically non-justifiable adaptation efforts.

A continuous learning is necessary to improve the reference model adaptation accordingly their adaptation preciseness, the ease-of-use of the adaptation mechanisms, and the cost of the adaptation process. Thus, adaptation controlling is the next logical step to improve the reference model adaptation. Within this paper we present an approach to controlling-enabled configurative information modeling for ES introduction. First, we
present related research in the next chapter. In chapter 2 we present the underlying research method. In section 3, we introduce the utilization of configurative reference modeling. Section 4 shows how controlling concepts are used to assure the quality of the configuration process. Section 5 provides conclusions and predicted further research.

1.2 Related Research

In order to enable a context-specific adaptation of information models, few conceptual modeling approaches have been developed in the near past:

Soffer et al. propose configurable information models in order to customize ES [2]. They use configurable, so-called Object-Process Diagrams that integrate process flows and data objects used within an ES. The configuration of these Object-Process Diagrams is performed by interpreting attributes that define the relation of diagram objects to different application scenarios. During ES customizing, the users have to specify their application context. Based on this, the attributes are interpreted, and the models are modified accordingly. The authors claim that it is necessary to connect model objects to a particular ES functionality in order to guarantee an easy application to the proposed configuration approach.

Rosemann and Van der Aalst propose a configurable reference modeling language that is based on Event Driven Process Chains (EPC) [4, 5]. The approach differs from that of Soffer et al. in so far as configurations are less predefined. It is based on semantic patterns in process models that describe dependencies of model elements on a semantic basis. Similarly to Soffer et al., the authors point out the need for connecting model elements according to ES functions in order to perform a model and ES configuration concurrently.

Becker et al. propose a configurable reference modeling language that is based on Event Driven Process Chains (EPC) [6] in order to comply with different modeling views that are needed for the integrated modeling of ES and surrounding business processes. In comparison to the approaches of Soffer et al. or Rosemann’s and Van der Aalst’s, Beck’s concept provides different configuration techniques that have different influences on the models. The approach provides a set of configuration mechanisms that are able to format modeling languages, models and model sections as well as model elements in order to fit to context specific requirements. Furthermore, the approach is not restricted to conceptual configurations of information models but allows also configurations of the graphical representation of models as well as the management of different languages and language-internal synonyms used in model element designations in different application scenarios. The introduction of ES is hence supported by automated model based and integrated ES and process configuration.

2 Research Methodology

Our contribution consists in providing configuration methods that target Enterprise System configuration. We follow Weick’s sense-making paradigm [7]. The relevance of the research topic was derived from requirements the authors were confronted with during information model-driven consulting projects. These were performed amongst others in the companies Deutsche Telekom AG [8] and Bayer Business Services GmbH [9] and the German Ministry of Defense. The requirements gathered within these consulting projects were balanced with already existing approaches to information model adaptation as briefly outlined in chapter 1.2, whereas a deficit of methodical support for this problem was identified.

As a basis for the scientific philosophical position of this paper a reference framework will be used that was presented by Becker, Niehaves and Knackstedt [8].

3 Configurative Reference Modeling

In order to provide efficient adaptation support, configurable reference models comprise rules which allow modifications of the original reference model depending on company or project individual determinations of configuration parameters.

3.1 Configuration parameters

When used in order to construct a specific information model, reference models have to be adapted to company or project specific issues. We distinguish the following parameters that a reference model configuration depends on:

- Business characteristics and their values represent sets of companies for which a reference model adaptation is to be performed. E. g., business characteristics are the performed business type or the trade level. Exemplary trade-related values of business types are warehousing, third-party-deal and central settlement; values of trade level are e.g. whole trade and retail trade [3].

- Perspectives represent information model requirements of different user groups. Perspectives are determined by the modeling purpose (e. g. ES customizing or process oriented organizational design), the organizational role of the actual user (e. g. manager, method expert or accounting clerk) and further influences like individual preferences according to the graphical design of the models [3].
3.2 Model projection

The most significant problem that results from a multiplicity of context specific variants is the need to manage possible redundancies inside the model itself. This leads to increased modeling and maintenance cost as well as the danger of inconsistencies within the model base. In the case of business process redesign, inconsistencies lead to a reduced applicability (e.g., for training), in the case of ES customizing, inconsistencies can even cause malfunctions of the software system.

In order to enable an efficient model construction and model maintenance, redundancies have to be overcome. A modeling language which enables users to avoid redundancies and to consider multiple variants within the model base is called configurative reference modeling (for the following cf. [3]). The approach is based on the concept of model projection. A configurable model that provides all relevant information for each variant contains constraints that determine to which variant each model element belongs. By this means redundancies are avoided and, simultaneously, the modeling of multiple application contexts comes possible. When a configuration is performed, each model element is hidden that does not belong to the selected variant.

3.3 Configuration Mechanisms Overview

For supporting the reference model user by model projection efficiently, it is beneficial to supply model projection based configuration mechanisms that have different impacts on the models. Therefore, configuration mechanisms are not exclusively proposed for the layer of modeling technique application (model layer), but also for the layer of their definition (meta-model layer). Configurative adaptations of meta-models act upon all models which were constructed in the corresponding modelling language, whereas adaptations on model layer only act upon specific models and model sections respectively.

In the following, we present examples of model configurations based on Event-Driven Process Chains with different configuration mechanisms (for a detailed conceptual introduction cf. [3]). We distinguish the following configuration mechanisms: Model Type Selection, Element Type Selection, Element Selection, Synonym Management and Representation Variation.

Within this paper, we focus on the Model and Element Type Selection as well as Element Selection, Synonym Management and Representation Variation do not play a crucial role within the adaptation controlling.

3.3.1 Model type selection

Model type selection considers the perspective-specific relevance of model types [10] which applies especially for organizational change projects. Model types represent the application types of modeling techniques, which can be combined in the context of reference modeling techniques, e.g., Event-Driven Process Chains (EPC) [11], Entity Relationship Model (ERM) [13], technical term models [10] or organizational charts. The configuration mechanism assigns perspectives to model types and so supports a coarse granular configuration of the model system. The model types, which are not relevant for the model user’s perspective, are hidden.

3.3.2 Element type selection

More detailed configuration rules existing on meta-model layer can be performed by using element type selection. Model types are characterized by the element types which are assigned to them and their relations among each other. In case of the EPC, these are e.g., functions, events, and resources. The configuration mechanism admits to build variants of model types by assigning element types to perspectives and, if necessary, by fading them out. E.g., variants of EPC differ in model element types which are annotatable to functions. Candidates for annotations are e.g., entity types, application systems and organizational units. The use of different types of the EPC takes place especially in reorganization projects. The model type variants are built by assigning element types to them which shall be faded out. Each model type variant is assigned to one or more perspectives.

![Fig. 1. Meta-meta model constructs for element type selection](image-url)

3.3.3 Element selection

The configuration mechanisms of model type selection and element type selection enable to hide element types perspective-specifically?.. Element selection, in contrast, permits the selection of single elements on model layer. There are different criteria for element selection which differ in the type of definition of the favored element subset.
Element selection via attributes: The creation of variants by element selection via attributes is achieved by analyzing the characteristics which are assigned to the reference model elements. An attribute, for example, can be intended for the functions of an EPC for the identification of its automation degree.

Element selection via logical terms: The element selection via perspective-specific selection of attribute values becomes inefficient as soon as the number of newly to be introduced attributes reaches a high level. Therefore the configuration mechanism of element selection via logical terms is additionally introduced, which enables to assign model elements directly to configuration parameter values by using a logical term. The annotation of an expression, e. g. “perspective (organizational design)”, to a model element flags the regarded element as relevant only for the perspective organizational design. The utilization of element selection via attributes can be recommended alternatively, only if already considered attributes of elements can be reused.

Secondly, the omission of model sections in business processes reengineering hides information that is not relevant. This omission can be recommended in the configuration mechanism of element selection via attributes. The configuration mechanism of element selection via logical terms allows omitting specific model sections which leads to exclude (or hide) specific parts of the ES functionality.

Fig. 2: Meta-meta-model constructs for element selection

Element selections are considered to be relevant both for ES customizing and BPR. First, element selection allows omitting specific model sections which leads to exclude (or hide) specific parts of the ES functionality. Secondly, the omission of model sections in business process reengineering hides information that is not relevant for specific user groups within the project, e. g. organizational designers in specific branch offices in which only parts of the whole business are performed.

Fig. 3. Grammar for element selection via logical terms

In case of element selection via logical terms, the relevance of a model element for a selected configuration depends on a logical term which is assigned to the element. In this context logical terms are understood as special attributes that can be assigned to model elements. They describe directly in which configuration the respective model element is available. Terms have to comply with a predefined grammar (cf. Fig. 3) for analysis purposes.

4 Adaptation Controlling

The configuration techniques presented above provide a proper mean to adapt models for different purposes but a successful adaptation requires an efficient adaptation process and the existence of suitable models to be adapted. The efficiency of an adaptation process can be measured in time and costs, that are needed to adapt the reference model to the aimed company-specific model. The goal of adaptation controlling is to minimize the necessary time and costs. We first introduce the enhanced reference model cycle that describes the necessary steps for a reference model adaptation controlling. Then we describe the mechanisms and meta-model enhancements that are necessary to perform an efficient model feedback and controlling.

4.1 Feedback Cycles in the Procedure Model

To minimize the effort that is necessary to adapt the reference model to a customer-specific model, the controlling phase strives for the goal to identify shortcomings and possible improvements of the reference model and its adaptation mechanisms.

The adaptation controlling phase is best positioned within the reference model cycle between the software usage and the requirements engineering (cf. Fig. 4), where first attempts of controlling exist with an unstructured feedback resulting from the experience with the software. The controlling phase can be structured in a sequence of three consecutive steps:

Data Gathering: The data that is gathered throughout the customization process can be differentiated in model data, adaptation and performance data. Model data mainly consists of the final adapted model resulting from the adaptation process. Adaptation data describes the process of adaptation and delivers information about the time that was spend on adapting the model, the order in which the adaptation was performed and e. g. the configuration steps that had to be reset due to misapprehension. Performance data provide information about how efficient the implementation process was performed and the resulting software works. Performance data is valuable for the reference model constructor (resp. software vendor) to evaluate new process variants. A fourth feedback is possible that describes the construction activities of the reference model. This data can be used to evaluate the reference model maintenance. This maintenance data is very similar to the adaptation data. Due to this similarity and the fact that we focus on the relation between customer and software vendor, we do not regard this feedback in this paper later on.
Data Merging: Data is sent from various customers containing different models, adaptation and performance data. To allow comparative analyses it is necessary to merge these data into one database (repository). In this approach we argue to integrate some of the performance data that are directly related to specific model elements (e.g. average cycle time of a process or of a process function) into the model data. This model has to be loaded into a repository consisting of the reference model and various customer-specific models (including different versions). The adaptation data that can not be directly connected to specific model elements have to be stored separately in an audit trail database.

Data Analysis: The data that has been gathered and merged from different customers is being analyzed in the third step. Target of this step is to identify shortcomings and improvements of the reference model and the adaptation mechanisms.

Fig. 4. Reference modeling cycle with controlling phase

4.2 Necessary Feedback Mechanisms and Model Enhancements

In the following we present the necessary enhancements within the conceptual model to enable the controlling and model maintenance. Therefore, as a key mechanism, it is necessary to retrieve the customer-model out of the model repository, which stores all customer-specific models together and integrated with the reference model. Using the configurative approach a customer-specific perspective has to be defined. The configuration mechanisms presented in chapter 3 do not allow selecting all the necessary model elements to show the customer-specific model. For instance attributes can not be assigned to a specific customer because they do not have attributes of their own. Therefore we introduce a new configuration mechanism, which selects model elements properties (cf. also in the following Fig. 5). Properties (as some kind of meta-data) should be assigned to all model elements. Thus, it is possible to assign a customer to an attribute or a constraint. In this case, we define properties as data that are not part of the model itself but are needed for maintenance reasons (e.g. date of creation, name of the creator, version ID etc.). To ensure the selection (Element Selection by Properties), the configuration parameter e.g. a customer perspective has to be assigned to the Property-Type.

Fig. 5: Meta-Meta-Model Enhancement for Element-Selection by Properties

To provide the data for controlling purposes the conceptual model has to be extended to describe the controlling relevant data and to distinguish the user-specific model. As an important precondition, every model element has to be identified regarding its source in the reference model. Therefore it is necessary to provide a valid identifier (Property Type: ID) for each model element (attributes, types, and constraints). This also allows identifying the model elements that has been omitted or additionally added. Therefore the modeling environment must support the generation of unique IDs throughout all modeling environments. The ID must not be mistaken for the unique ID (Primary Key) provided by the database to distinguish the table entries. The ID, described in this paragraph, identifies the model element that can exist in various variants throughout different customers. All these variants share the same ID. Thus, for the ES vendor it has a grouping character, grouping all variants.

All Elements, uploaded by the customer, have to be recognized in the customer’s perspective. This can be achieved during the upload phase where the customer identifies himself by initiating the upload session. Every uploaded model element is assigned to the customer (Property Type: Customer). This also allows having customer-specific attribute instances integrated in one model element. Therefore the Meta-Model of the
modeling language has to be altered by being less restrictive e.g. allowing several attribute instances for one attribute type or multiple outgoing edges within an EPC-Model. Nevertheless, the customer is only provided with single and unique attributes. Hence, consistency is assured by keeping the original modeling language on the customer side so that only one instance is possible.

The nature of a cycle is that every step is performed not only once. Therefore the model must ensure that the customer is able to upload an advanced model later on. This requires properties for versioning. Every model element that has been uploaded is assigned to customer-specific version ID, identifying the model elements that are part of a version. Model elements that have not been changed are additionally assigned to the new version.

Performance data can be integrated into the customer-specific model as attributes in the customer’s adapted model. But this is only possible for data that is standardized and comparable. To provide performance data e.g. of single process instances an audit trail would be the best solution. To provide a reference from the performance data to the model element, to which it is connected, the audit trail has to provide a field for this assignment. The reference data that can be provided with the model should give performance information like the average cycle time for a process or a process function. Within the conceptual model on the meta-model layer the element attribute has to be specialized in the types (e.g. cycle time) that should be provided by the customer. The resulting meta-model is depicted in Fig. 6.

![Fig. 6. Meta Model Extension for the Data Gathering](image)

### 5 Conclusions and Further Research

It has been shown how the adaptation controlling of conceptual reference models to specific application contexts can be supported more comprehensively than it was possible before. Therefore, recommendations for the construction of reference modeling languages in terms of configuration mechanisms have been formulated.

The conceptual specifications of the configuration mechanisms can be used as a basis for the implementation of modeling tools that are able to support configurable reference modeling. Because of the extensive complexity of configurable reference models that results from the integrated configuration rules, the support by modeling tools is a necessary precondition for the application of configurable reference modeling in practice. Handling configurable reference models with common draft applications is – in case of realistic project volumes – considered to be impossible due to acceptance and cost barriers. Hence our further research and development work focuses at implementing configurable reference modeling as a plug-in for a well-known modeling tool. Additionally, the plug-in will be used as a basis for the evaluation of the proposed configuration mechanisms in practical fieldwork.

### References