ADAPTIVE MODELING OF INTER-ORGANIZATIONAL BUSINESS PROCESSES IN TELECOMMUNICATION SECTOR

[2]Silvio Tschapke, [1]Iris Braun, [2]Rene Fleischhauer, [2]Michael Stollberg
[1]Technische Universität Dresden, Faculty of Computer Science, Chair for Computer Networks
01062 Dresden, Germany
Email: iris.braun@tu-dresden.de
[2]SAP Research CEC Dresden
Chemnitzer Str. 48, 01187 Dresden

Abstract – Inter-organizational business processes are becoming increasingly prevalent in the telecommunication sector. Numerous partners such as Telecommunication vendors, Internet service provider (ISP) or Cloud provider collaborate across organizational borders in a highly dynamic environment. Seamless integration and efficient management of data and information is indispensable for such processes characterized by permanent re-planning. In short, the right information has to be available at the right time, in the right place. Although appropriate information management is necessary, the technological support is still limited. The paper presents an adaptive modeling approach based on Configurable Collaboration Artifacts as key elements for representing collaborative business processes. The solution approach extends the concept of Business Artifacts by mechanisms of runtime variation.

I. Introduction

The exchange of data and process information between telecommunication companies often requires considerable manual effort. Research approaches related to business process management are aiming to solve these problems, however existing solutions lack of adequate agility and transparency of the overall process. Therefore, additional management capabilities must be provided by a solution that builds on top of existing technologies and infrastructures. The problem thereby is to modify certain parts of the process, without interrupting its execution.

The paper addresses the research question of how the modeling of inter-organizational business processes can be improved to get a more flexible, and scalable system, which in addition provides sufficient information transparency to execute the process efficiently. With regard to this research question, the solution approach introduces an adaptive modeling approach, based on Configurable Collaboration Artifacts. The main idea is to extend Business Artifacts with the ability of (re-)configuration with pre-defined, but variable Features. This approach enforces modularization and as a consequence increases reuse and flexibility.

II. Concept of Configurable Collaboration Artifacts

Our solution approach is about modularization and configuration of business processes to concrete needs of an organization for a specific scenario. [Rosa2008] describes the idea of process configuration by asking the question: „how to model business processes that are similar to one another in many ways, yet differ in some other ways from one organization, project, or industry to another“. Answering this question is the key to increase flexibility, and as a consequence to reduce the need for change. The implementation of configurative capabilities however requires specific design constructs in order to support the construction of process variants by reusable parts of the process model.

The central idea of the solution approach is to extend the artifact-centric modeling paradigm with configurative such capabilities. Instead of declaring artifacts with a fixed data and life-cycle model, we make the definition and the runtime behavior of artifacts more flexible. Configurable Collaboration Artifacts are based on the approach of Business Artifacts [Hul08] and on the related GSM meta-model [Hul+11]. Business Artifact configuration allows adapting a whole business process to changing needs. Process flexibility can be realized using different patterns. Our approach increases flexibility by design and by under-specification. For a concrete Business Artifact this means that only the fixed parts are declared at design time, whereby the binding of variable parts is postponed (see Figure 1).

The main motivation of Business Artifacts holds for Collaboration Artifacts as well: each domain object which is subject for collaboration holds all the information required to fulfill its operational goal. This time however, this artifact is configured to organization-/scenario-specific requirements in the first place. This approach combines the benefits of both artifact-centric modeling and configurable business process modeling. First, we keep the design focus on the data, which is of primary interest for many processes. Beyond that, an artifact provides a global perspective across organizational boundaries by a fundamental combination of its data- and life-cycle model. The question is: “which data drives the business?” Second, we increase flexibility of Business Artifacts by applying configurative techniques. Therefore, the technical architect has to identify the parts of the process, which are likely to change. Instead of defining as many and as complex artifact types as possible, we keep the design slim and flexible, to support multiple scenarios without the need to interrupt or re-design a running process. The challenge is how to modularize an artifact system in an appropriate way.

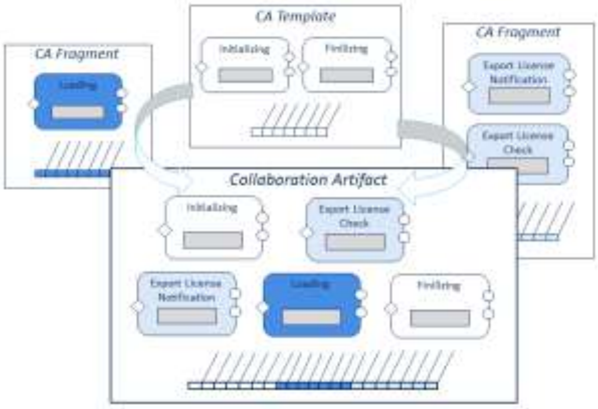

Figure 1 - Configuration of Collaboration Artifacts

The self-contained parts constitute all information required to realize an aspect of a related artifact. This relation between artifacts and fragment is described by the “template-and-hook” paradigm [Pre95], which allows describing both the common and the variable parts of a software component. A collaboration artifact template represents the common part of the raw data element. It contains all functionality, which is required for collaboration, and which is equal among all organizations and all possible process variations. Common attributes are an artifact template name as a unique classifier of this type and meta-data for management and monitoring purposes. Each template has a declarative life-cycle model consisting of guards, stages, and milestones to define the constraints of the corresponding operational goal. Since an artifact template is an incomplete skeleton, these attributes or the set of life-cycle stages might be empty. This leads to the central point of artifact templates, i.e. the set fragments that are bound to the hooks and influence its behavior. Such fragments are integrated transparently into an artifact, which is finally executed to realize the business process. The integration of additional functionality is done via the configuration interface of template hooks.

A collaboration artifact fragment represents a chunk of functionality which can be added to an artifact. This concept is similar to features of product lines or to aspects in aspect-oriented programming. Features have interdependencies which are expressed in feature trees and variability diagrams. Such relations are required for artifact extension as well. “Loading requires unloading” is one example of a possible relation. Since artifact-centric modeling requires the fundamental combination of data and process models, this holds for the introduced fragments as well. This means each fragment is defined by a concrete data attributes and an internal life-cycle model which can be expressed by the GSM meta-model. In contrast to an artifact template, a fragment is not extensible. This limitation is important to keep the variation space and the overall model manageable. However, since warehousing requires different document types or process constraints from one organization to another, fragments are still variable. Whereas artifact templates are extensible via extension points, a fragment is variable via variation points and parameterization.

III. Setting up and executing a Collaboration Artifact System

The concept of configurable collaboration artifacts enables the declaration of a flexible meta-model for building artifact systems. Flexibility by configuration however requires technical management support. A traditional design process consists of: 1) Analysis, 2) Design, 3) Deployment, and 4) Execution. We adapt this cyclic process by dividing the design phase into: 2a) Design of Artifact / Feature Types and 2b) Configuration. Whereby artifact template types and artifact fragments must be designed a-priori, the actual configuration is executed at runtime. One configuration step consists of: first, the parameterization of a fragment, and second, of the binding of this fragment to a template. This procedure is repeated until the template is enriched with all the required functionalities. The Configuration Engine is responsible for the creation and configuration of Collaboration Artifacts. This process is triggered by an incoming / imported description of a concrete business process scenario. By means of a repository of pre-defined templates, fragments, and configuration rules, an artifact is configured and finally deployed to the Collaboration Artifact Manager.

Artifact instantiation and its subsequent configuration result in an artifact system which represents an executable business process model. The challenge in process modeling is in determining whether the model exhibits a certain behavior or not. Process verification can greatly improve the reliability of executable business process models [Wyn+09] and therefore is of great interest of current research in artifact-centric modeling [ACSI]. We are interested in guaranteeing certain properties after each configuration step, which primarily covers the reachability of operational goals.

The Collaboration Artifact Manager is the central component. It fulfills the role of the runtime container for a set of Collaboration Artifacts, which are intended to represent business related data and activities. At runtime, the instantiated and configured collaboration artifacts behave in the same way as Business Artifacts, which means that the message-based interaction of several artifacts realizes the business process. The Artifact Manager controls this interaction, as well as the runtime states and create-read-update-delete operations of single artifact instances. Message-based interaction has the advantage that the artifacts are dynamically exchangeable at runtime. This is important aspect for the required flexibility. For minor changes, whereby no complete artifacts have to be exchanged, the Artifact Manager requests the Configuration Engine for (re-)configuration.

IV. Conclusion and future work

The developed solution approach allows efficient management of constantly changing business processes, as well as provides real-time notification support for process deviations. The central scientific contribution is in extending the artifact-modeling approach, by introducing concepts for process configuration. This provides more flexibility by varying and extending single artifacts. Furthermore, such modularization keeps the design slim and the set of artifact types manageable. This solution approach can be used as an adaptive design-, and execution model for business collaboration management systems. The prototypical implementation demonstrates technical feasibility.

However, not all discussed concepts are now realized in code. A possible extension would be the enhancement of the Collaboration Manager with fully automated support for artifact configuration. This process of artifact instantiation could be further enhanced and controlled by additional configuration rules. Another possible extension would be the integration of persistence and of security concepts and frameworks.

V. References

[Rosa08] *Marcello La Rosa and Marlon Dumas.* “Configurable Process Models : How to Adopt Standard Practices in Your Own Way?” In: BPTrends (2008), pp. 1–10.

[Hul08] *Richard Hull*. “Artifact-centric business process models: Brief survey of research results and challenges”. In: On the Move to Meaningful Internet Systems: OTM 2008 5332 (2008), pp. 1152–1163.

[Hul+11] *Richard Hull et al.* “Business Artifacts with Guard-Stage-Milestone Lifecycles : Managing Artifact Interactions with Conditions and Events”. In: Proceedings of the 5th ACM international conference on Distributed event-based system. 2011, pp. 51–62.

[Pre95] *Wolfgang Pree*. Design Patterns for Object-Oriented Software Development. Addison Wesley Longman, 1995.

[Wyn+09] *M.T. Wynn et al.* “Business process verification: finally a reality!” In: Business Process Management Journal 15.1 (2009), pp. 74–92.