

INTEGRATING PROCESS AND SERVICES THROUGH META-MODELS

ABSTRACT

The service conceptualization within a SOA/BPM methodology reduces the gap between the business process life cycle and their executable versions. The SOA/BMP methodology supporting this article, proposes a set of steps; interactions between such steps require the definition of mechanisms that can be formalized. This paper is based on the use of meta-models as a means to define the language syntax of each stage and unambiguously transformation rules between them as a step towards automatic transformations. In previous work a service meta-model conceptualization integrated with the BPMN meta-model was proposed. The SCA standard (Service Component Architecture) describes how to create and combine components. In particular, this paper proposes a meta-model integration between our service meta-model conceptualization and the SCA meta-model.

KEYWORDS

SOA, BPM, MDD, Meta-model. SCA. Components.

1. INTRODUCTION

The integrated vision of service orientation and process orientation leads to define a methodological framework that orders the concepts and guides the lifecycle of business processes in order to obtain executable processes. In this regard, the work developed in [2] defines a model of integration between SOA (Service Oriented Architecture) [13] and BPM (Business Process Management) [16]. This model proposes a general and comprehensive methodology to develop projects with SOA and BPM approach. This methodological proposal, consisting of eight stages - 1) Stage of Organization and Strategic Plan, 2) Identification Phase and Requirements Specification with process approach, 3) Business Modeling Phase, 4) Process Modeling Phase, 5) Service Modeling Phase, 6) Definition of Components Phase, 7) Components Implementation Phase and 8) Management and Monitoring Phase (see Figure 1) outlines a new global vision that identifies each stage and its interaction with the next, covering both the life cycle of business processes such as software in a unified manner. However, some of the interactions between these stages still need to be deepened and precisely defined. The aim of our research is to improve the execution of each stage and create more robust products that provide feedback to the cycle.

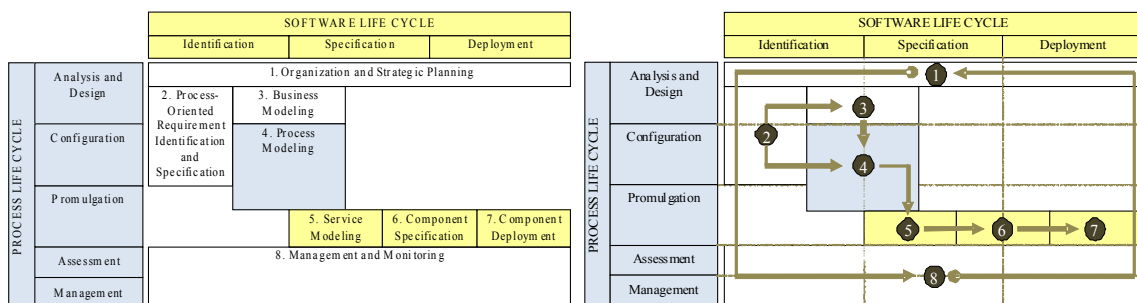


Figure 1. Phases of the SOA/BPM integrative methodology [1]

One of the most obvious and core interactions in the methodology proposed in [2] is the process with the services. For this reason this paper focuses on reviewing and improving interaction between: 1) processes -

modeled as a refinement step from the business model and 2) services - conceptualized to respond to the processes activities. Conceptualized services can be considered as new business services built to perform the process activities, or as business functions of existing applications to be reused as services.

Also, in addition to identifying the points of interaction between relevant stages of the methodology, this paper seeks to achieve the formalization of the products of each stage through the mechanism of metamodeling. Defining a metamodel brings precision to the language and enables to build tools for editing model elements, facilitating the actual testing of the proposed methodology. Metamodeling is a mechanism to formally define the syntax of modeling languages unambiguously which allows different tools to interpret the models [12].

In this framework, we have defined- in [3]- a metamodel for service conceptualization from the processes and activities, based on an adaptation of SOAF metamodel (Service Oriented Architecture Framework) [7] and combined with the BPMN metamodel [8] [4]. The proposed metamodel is instantiated with a concrete example and a prototype of a graphical editor that provides the concrete syntax of it was developed, allowing to have a tool for editing services from the processes activities.

From the metamodel of service conceptualization proposed in [3], we obtain a tool for modeling services. The next interaction step (according to [2]) in this modeling is the definition of software components to be combined to absorb the flow execution that the process marks.

Architecture SCA (Service Component Architecture) defines a general approach for creating components and describes how they work together [13]. In particular, the SCA specifications define how to create and how to combine components into complete applications beyond the component technology used.

This paper provides an integration between the service conceptualization metamodel P2S (ProcessToService) proposed in [3] and the SCA metamodel as a step of interaction between service modeling stages - stage 5 and stage 6 - of the methodology. This integration also defines a set of transformation rules between elements of each language which is presented as a concept association table that will represent in future works the conditions to write transformation functions.

This integration of metamodels contributes to a global objective which is to produce a new version of the integrating methodology SOA / BPM, model-driven and maximizing the automation of tasks based on model transformations, thus formalizing the steps of interaction between each stage of the methodology.

Specifically, the contribution of this paper is to formalize another interaction step between service modeling and component modeling in the context of integrated methodology SOA / BPM proposed in [2], in order to automate the production of a model of software components aligned with the production of business processes models, through services modeling.

The paper is organized as follows. In Section 2, we introduce briefly the integrated SOA-BPM methodology behind this work and it is illustrated through a case study. Section 3 shows the services conceptualization metamodel P2S as a step towards the stage of definition of components that will be resolved through the SCA standard, the definition of components is detailed in Section 4. In Section 5, we propose a form of integration between SCA metamodel and service conceptualization metamodel P2S. Finally, we show the conclusions.

2. METHODOLOGY INTEGRATING SOA / BPM APPLIED TO A CASE STUDY

Recently, the BPM proposal has gained considerable attention both in academia and industry software. BPM is a strategy to manage and improve business performance by optimizing their processes through modeling, execution and performance measurement within a cycle of continuous improvement.

SOA is a different approach for the design and construction of flexible and adaptable systems that assist the development of systems in a dynamic business environment. While it is not a new concept it is being widely studied in academia and used in practice.

In a SOA environment, services can be shared and reused in multiple business processes. The result is a highly adaptable environment, with a greater return on investment in application development, better integration and faster systems deployment.

In previous work [2] a methodology has been developed in order to conceive a model for integrating applications within an organization, so as to align the processes that define its performance with the services

that support the functionality. This methodology places a strong emphasis on the identification of requirements with a process approach. It also has a business modeling stage, prior to the modeling of processes and services. From modeling services, succeed two stages of design and components implementation, respectively. Finally, there are two global stages: an organizational and strategic plan at the beginning and subsequent management and monitoring stage, which cross and involve the internal phases.

As envisioned in [2], the modeling stage of the business produces a map of processes that with the identified and specified requirements and the business use cases, provide the basis for modeling business processes.

From the business model, business processes modeling is a step of refinement, where processes are enriched with functional aspects by defining system use cases and describing the information objects. This refinement step enables to decompose the process in tasks and define roles and flows that and provide textual documentation for the description of the business processes (BPD).

Once the process model is obtained, we have the ground for identifying the elements (services) that will perform business processes, the latter becoming consumers of such services.

The following sections introduce a case study to demonstrate the application of some of the stages.

2.1 Case study

In this section we present a case study corresponding to the modeling process that develops within a service center of a transport company, among drivers who apply for vehicle repair and the workshop that carries out the task. The process generates a work order (OT) when the driver arrives with transportation to repair. For every activity of repair, if it can be planned, the allocation of a box and a mechanic for each item in the OT is programmed. Service is performed calculating the actual cost and material used and then the OT is closed.

In [3] we describe the application of P2S to a case study which uses the metamodel of conceptualization, mapping the activities of a process as services of the service model. It starts with a model in BPMN which identifies the activities of the process. From them we obtain a set of services that are documented with laced circles as determined by the proposed methodology. The way of linking services from activities, is supported by our metamodel, using the reference "perform" between activities and services.

In this paper we apply on this example, the interaction step of the service conceptualization metamodel to the component modeling using integration with SCA metamodel.

2.2 Process modeling and classification

The heart of the business process modeling are the processes themselves. Applying different levels of abstraction, there are three related concepts: 1) Processes, 2) Sub-processes and 3) Tasks. A process is a network of "things to do." A Sub-processes is a process in itself, whose functionality is part of a larger process. The lower-level process that cannot be decomposed is considered a Task.

As mentioned in [3] this step is represented by BPMN through the rotational elements identified in the metamodel. Figure 2 shows the model in BPMN of the process set forth as a case study, which was built with the Eclipse BPMN editor.

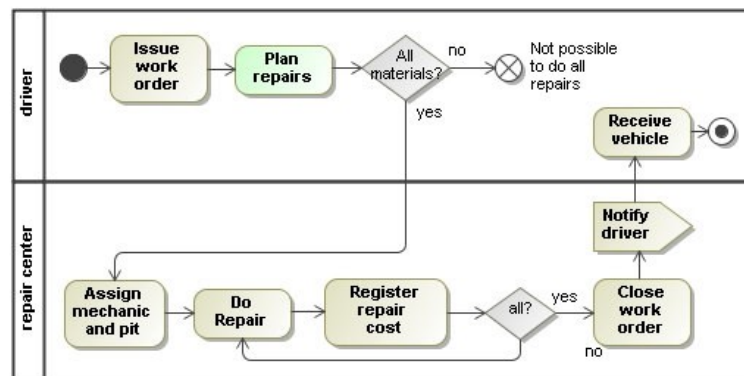


Figura 2. Work Order Process

2.3 Services modeling and classification

Services may be, within the paradigm of object orientation, as objects that provide more information and less coupling as usually they are not associated, their behavior is triggered by the meta-information they have, concerning service contracts and execution context. Because of their similarity to the objects they can be modeled in terms of interfaces.

Inspired by UML classes, Erl defines in [15], the symbol of service as a circle comprised of two areas: one for the service name and one for the capabilities.

A simple service provides a collection of capabilities that are grouped by functional context established by the service.

The ability of a service represents the specific function thereof through which it is invoked. They are expressed within the service contract.

Service contracts indicate the interfaces that provide the services - their operations and their parameters, favoring the documentation of services and the resulting composition to extend the functionality.

The granularity of a service communicates the level of detail associated with its functional scope. There is no single criterion that indicates the level of granularity for a service, but the design principles and classifications of services adopted impact directly on the level of granularity.

The classification proposed in our metamodel takes into account services on: business, organization and infrastructure.

- *Business services* have the functional scope of business entities, are of high reusability and are agnostic to many business processes.

- *Entity services* have functional limitations associated with classes within a class model. They have less reuse potential and are composed with other services.

- *Cross-cutting services* are infrastructure services that solve aspects of infrastructure or technology services. As an example, a centralized service reporting exception conditions or a service that establishes a connection to a relational database in a generic way, are cross-cutting services.

3. SERVICE CONCEPTUALIZATION METAMODEL

The service conceptualization metamodel SOAF can adapt and focus on aspects of the processes specifically taken as a set of activities, each one of them is the performance of a service seen as a functionality abstraction.

That is how we define a metamodel for conceptualizing these services based on the SOAF conceptualization metamodel [7] and the BPMN metamodel defined in [8] which was inspired by the proposed by Eclipse [9] to model business processes. The proposed metamodel, called P2S, conceptualizes services as a generalization of the internal components and external services. It is built as instance of standard language MOF (Meta Object Facility) [11], which constitutes the most abstract layer in Architecture 4 modeling layers of the OMG [10] [12].

The process of obtaining services from the activities of a business process is one of the key points in the interaction between processes and services. The P2S metamodel determines that the metaclass Service "realizes" an Activity. This aspect is implemented using a combination top-down and bottom-up. That is, when applied top-down analysis to identify the activities, we apply a bottom-up analysis for the functional components that respond to business activities under the premise of providing the information to be managed, the operations for handling and business rules that govern such manipulation.

After this bottom-up analysis, are determined which activities are performed by each identified service, which, in turn, will be classified according to the generalization of the metamodel metaclasses.

This metamodel was instantiated with the case study and a graphical editor was built [3] using Eclipse plugins included in the Eclipse Modeling project [5],[14],[6].

In the next section we present the definition phase component of the methodology proposed in [2], whose modeling is based on the SCA metamodel and we propose an integration between this metamodel and our of service conceptualization metamodel P2S.

4. COMPONENTS DEFINITIONS STAGE IN THE METHODOLOGY

Under the proposed methodology [2], to maintain their independence, services should encapsulate logic within a context that can be a task, a business entity or some other grouping. In order for services to be orchestrated and for them to deploy their operation logic, they must intervene in the execution of the business activities, for which they must be able to establish relations with those who want use them.

Following the period of service modeling and their identification, comes the stage of components definition, is packaging of services such as coarse-grained components, to meet the functional and nonfunctional requirements identified in the initial stages and framed in the business processes that require them.

The definition phase component is understood as a set of bundled services that support a single entry and may have multiple outputs.

4.1 SCA and its metamodel

SCA (Service Component Architecture) defines a general approach for creating components and describing how they work together [13]. Now they are an OASIS standard [1], originally created by a group of vendors including BEA, IBM, Oracle and SAP among others.

The SCA specifications define how to create and combine components in complete applications. Components can be constructed with JAVA and other languages using programming models based in SCA or can be constructed with other technologies like BPEL (Business Process Execution Language) or the framework Spring. Beyond component technology used, SCA defines a common mechanism to specify way it combines the components in applications.

Figure 3 shows a simplified metamodel SCA standard defined in [13]. There we observe that: a *Composite* contains *Reference*, *Component*, *Wire* and *Service*. In turn, *Component* contains *ComponentReference* and *ComponentService*. *Reference* and *ComponentReference* are subclasses of *BaseReference* and *ComponentService* and *Service* are subclasses of *BaseService*, which contains *Operation*. *Reference* and *Wire* relate with *ComponentReference*. In turn, *Service* and *Wire* relate with *ComponentService*. *Component* contains *PropertyValue*, while *ComponentType* contains *Property*.

5. INTEGRATION OF SCA METAMODEL WITH THE SERVICE CONCEPTUALIZATION METAMODEL P2S

The SCA metamodel described in the previous section defines a language for specifying a component model that partially covers the relevant aspects of the language. Since this is a simplified metamodel, it is possible that some elements are not complete, but this simplification is sufficient to analyze the integration with the service conceptualization metamodel (P2S).

One of the most obvious points when analyzing an integration between SCA and P2S comes from the fact that *Component* in P2S is a specialization of *Service*, whereas in SCA, both *Component* and *Service* are part of *Composite*. That is, *Service* does not share the same level of abstraction in both metamodels.

Beyond this point, we can observe that the SCA standard allows, through its metamodel, to define a specification language at stage 6 of the methodology (Definition of Components). The interaction between this stage and the previous one (stage 5 – Services Modeling) may be solved by instantiating both metamodels but disambiguating the concept of *Service*.

The most obvious way to disambiguate this element is to define that *Service* in P2S, actually represents a model of service and not the service itself. Is a conceptual service that may be a component or an external service (such as described in P2S). In this regard, the *Service* of P2S metaclass is then named as *ConceptualService* and so both metamodels are linked through the metaclass *Component*.

In turn, the *Component* metaclass used as a means of connection between the two metamodels, can also be renamed to *ConceptualComponent* in P2S and is related to *Component* in SCA. Figure 4 presents a shortened version of the proposed integration of metamodels.

In light of the above, Table 1 presents a set of rules linking each metamodel metaclass. These rules are a first step in defining transformations between metamodels, which will be subject of future research.

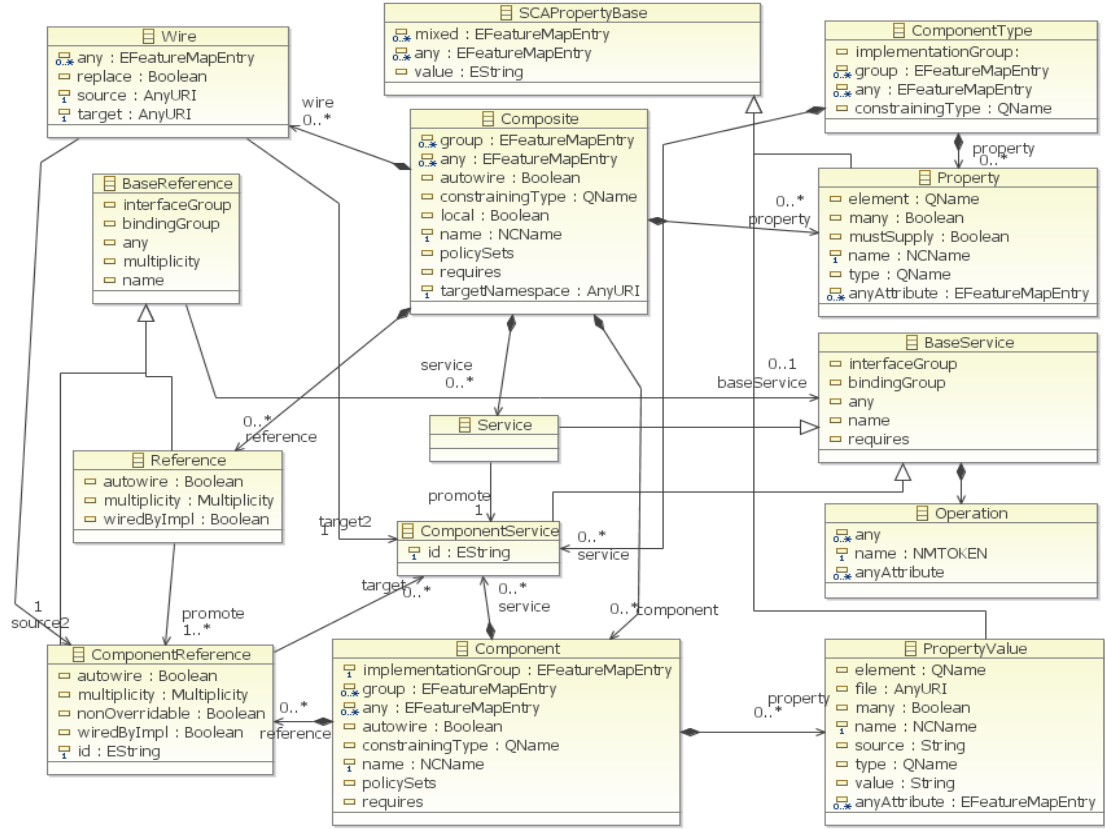


Figure 3. Simplified SCA Meta-Model

TABLE I. INTEGRATION RULES BETWEEN METAMODELS

Metaclasses BPMN	Metaclasses P2S	Metaclasses SCA
Process Model	Process	Composite
Pool	ProcessService	Component
Connection	Contrato	Wire
Artifact	TypedElement	Reference
Activity	ConcpetualService	Component Service

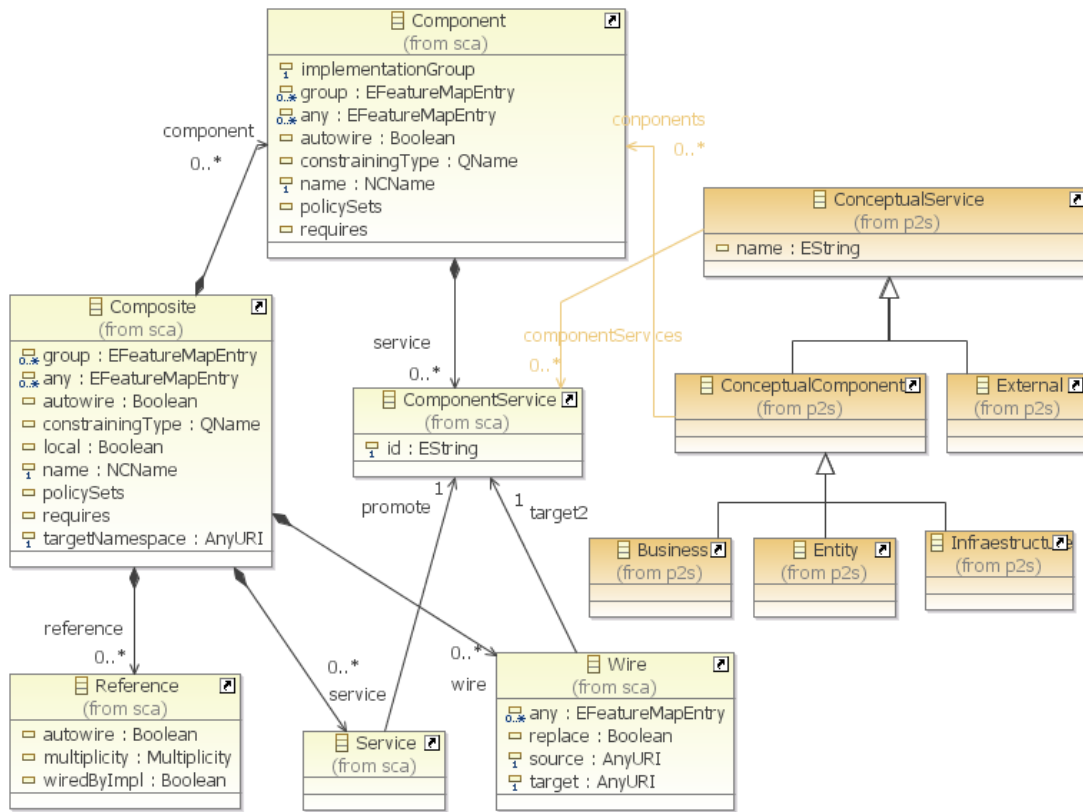


Figure 4. Linking SCA metamodel with P2S

6. CONCLUSIONS

This paper is a step towards improving the SOA / BPM integrating methodology proposed in [2] and enriched in [3], by service conceptualization modeled in a stage of this methodology.

This view contributes to a more global objective which is to produce a new version, model-driven, of the SOA / BPM integrating methodology, maximizing the automation of tasks based on model transformation. The objective is to achieve a formalization of transformation and interaction steps between each stage of the methodology.

Also we expected to contribute to a new methodological approach that achieves a balanced combination of alternate use of top-down development techniques for business process development and bottom-up techniques for service development.

In this work we present an integration between the service conceptualization metamodel presented and implemented in [3], with the SCA metamodel for the purpose of having a modeling language syntax for the next stage of the methodology that consists on obtaining a components model from a service model. This integration defines a set of rules that will be used in future work to automatically obtain a component model, based on modeled services.

REFERENCES

- [1] Advancing open standards for the information society. <http://www.oasis-open.org/>
- [2] Bazán P. 2010. "Un modelo de integrabilidad con SOA y BPM". Tesis de Maestría en Redes de Datos. Facultad de Informática. Universidad Nacional de La Plata.

- [3] Bazan P., Perez G, Giandini R., Diaz J. 2011. "Process - service interaction using an SOA-BPM methodology". Publicado en XXX Conferencia Internacional de la Sociedad Chilena de Ciencia de la Computación (SCCC'2011). Curico, Chile.
- [4] Business Process Model and Notation (BPMN) 2.0, Beta 1, OMG, May 2009
- [5] Eclipse Modeling Framework EMF. <http://www.eclipse.org/modeling/emf/>
- [6] Eclipse EuGENia. www.eclipse.org/gmt/epsilon/doc/eugenia/.
- [7] Erradi, A., Anand, S., Kulkarni, 2006. SOAF: An Architectural Framework for Service Definition and Realization, IEEE Int. Conf. on Services Computing, pp. 151-158.
- [8] Roxana Giandini, Gabriela Pérez, Claudia Pons. 2010. Un lenguaje de Transformación específico para Modelos de Proceso del Negocio. XXXVI Conferencia Latinoamericana de Informática (CLEI 2010). 18 al 22 de Octubre de 2010, Asunción, Paraguay.
- [9] S. Kühne, H. Kern, V. Gruhn, and R. Laue. 2010. "Business process modeling with continuous validation" Journal of Software Maintenance and Evolution: Research and Practice. Volume 22, Issue 6-7, pages 547-566, 2010. DOI: 10.1002/smr.517
- [10] Object Management Group (OMG), <http://www.omg.org>
- [11] OMG/MOF Meta Object Facility (MOF) 2.0. OMG Adopted Specification. October 2003. <http://www.omg.org>[11] <http://www.eclipse.org/stp/sca/>
- [12] C. Pons, R. Giandini, G. Pérez. 2010. "Desarrollo de Software Dirigido por Modelos. Conceptos teóricos y su aplicación práctica". EDULP & McGraw-Hill Educación. ISBN: 978-950-34-0630-4. Capítulo 4.
- [13] SCA Assembly Model Specification 1.1, Open SOA Collaboration, Mar 2009.
- [14] The Eclipse Project. Home Page. Copyright IBM Corp, 2000.
- [15] Erl, Thomas. 2007. "SOA Principles of Service Design". Prentice Hall. ISBN-13: 9780132344821. Pag.25-119
- [16] Weske Mathias, 2008. "Business Process Management: Concepts, Languages, Architectures". Springer, Pag 3-67. ISBN 978-3-540-73521-2.