A Service-Oriented View on Business Processes and Supporting Applications

Sebastian Adam, Matthias Naab, and Marcus Trapp

Fraunhofer IESE, Fraunhofer Platz 1, 67663 Kaiserslautern
{sebastian.adam,matthias.naab,marcus.trapp}@iese.fraunhofer.de

Abstract. Even if SOA has received much attention, there is still no common definition of what a SOA is or what a SOA should provide for business. In this paper, we have therefore introduced a conceptual model on service orientation that explains the impact of service orientation on business processes and supporting applications. We consider this consolidation as an essential step for establishing methods for a better business IT alignment as well as more systematic and integrated business and software engineering in the context of service-oriented enterprises.

1 Motivation

Service orientation and especially service-oriented architecture (SOA) [1][2][3][4][5] have received much attention in research and industry. Nevertheless, there is still no common definition of what a SOA is or what a SOA should provide. Rather, many different definitions can be currently found in the literature.

A traditional and still dominant perspective considers SOA just from a technical viewpoint addressing web service, WS-* protocols and the like [6]. A similar but rather conceptual view highlights SOA as an architecture style for managing heterogeneous application landscape [1]. Finally, in the business community, SOA is increasingly considered as an enterprise architecture or even management concept that promises structuring the business in order to remain more flexible.

While all these specialized definitions and views have their eligibility, they tend to neglect that their might be a more holistic view on SOA. Thus, the multitude of existing viewpoints, which originate from a limited perception, makes it currently hard to get a consolidated picture that could enable a more holistic management of business and IT following the service-oriented paradigm.

In this paper, we therefore claim that all different views on SOA should be consolidated and aligned with traditional business process concepts via a service-oriented business process perspective. Such a perspective could facilitate the explanation of how business issues, application landscapes and technical details are related and how one aspect can influence another. We consider this consolidation as an essential step for establishing a better business IT alignment as well as more systematic and integrated business and software engineering in the context of service-oriented enterprises.

I. Bider et al. (Eds.): BPMDS 2010 and EMMSAD 2010, LNBIP 50, pp. 39–48, 2010.

As a basis for this aim, this paper therefore introduces a conceptual view on services in the context of business processes and supporting applications that clarifies how the notion of service orientation fits to process-related issues both from an organizational and from a technical perspective. To make that happen, we introduce a service classification and explain how different types of services are used in, respectively produced by, business processes. Furthermore, we clarify the different roles involved in the provision and consumption of services, and we enhance the traditional provider-consumer-broker triangle [1] in this regard. Finally, we deal with the question how service orientation coheres with electronic channels such as the internet. An illustrating example in section 3 shows how our conceptualization can be applied in order to express service orientation in real business settings. The paper closes with a discussion on how we are going to use the conceptualization for developing integrated engineering methods for aligning business and software engineering.

Of course, our idea of consolidating the multitude of views on SOA has been inspired by existing work that also goes towards this direction. In the SOA reference architecture of OASIS [7], for instance, a comprehensive model of many SOA aspects has already been given. However, the business process aspects are rarely covered in this model and only services as such are in the focus of attention. Therefore, a further source of inspiration was the model proposed in [8]. It provides a good coverage of business processes and service concepts and introduces a clear separation of business and IT. We complement the ideas introduced there with more concepts related to the roles involved in typical SOA settings and include more detailed information on the relationship of business services and software services. In particular, beyond the views covered in this paper, our entire conceptual model covers further aspects related to engineering questions and the realization for SOA-based IT-systems.

2 Service Orientation in Business Processes

In this section, the impact on service orientation on business processes is discussed according to our conceptual model we developed in this regard. The section is subdivided as follows. In the first sub section, we explain which types of services can be distinguished. In section 2.2., we then present how the different types of services are aligned with traditional business process elements. In section 2.3., different channels over which services can be requested and consumed are introduced. Finally, in section 2.4., we present different service consumer and service provider roles.

2.1 Service Classification

According to our conceptual model (which is presented using UML in the following), a service is basically "*a clearly defined unit of work, which is provided by a provider and consumed by a consumer and bears an immaterial value for the consumer*". However, in research and industry, the term "service" is often used in a homonymic way expressing each kind of value provided from one party to another without distinguishing the purpose of the service or the type of the provider. It is therefore important to differentiate the types of services that have all been covered under the single term "service" so far.

In our conceptual model, we therefore consider a service as an abstract term that is specialized into business services and (specific) software services (see Figure 1). A business services is *"a service provided by a business unit that is of immaterial value for at least one other business unit"*. Transporting goods from one location to another is an example for a business service in the logistics domain. However, business services can also be for internal purposes (hence, provided to in-house business units) and not only external business services that address external consumers.

In contrast to a business service, a software service is *"a service that is provided by a software system and used by the same or another software system"*. Software service remains also an abstract term and is therefore specialized into either interaction service, function service, or infrastructure services depending on its concrete purpose. While an interaction service is *"a software service controlling a long-lasting interaction among software systems and users"*, a function service is *"a software service providing a self-contained functionality, which can be requested from a software system."* What we call function service here is often called simply service, application service, or web service in other terminologies.

A function service can be either atomic or molecular. While an atomic function service is a function service that is not realized by composition of other function services, a molecular function service is a function service that is realized by such a composition.

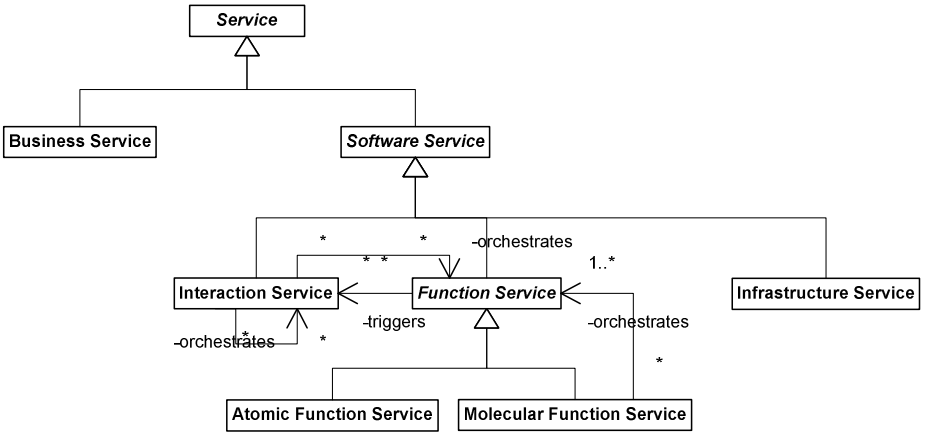

Fig. 1. Service Classification

Function services may trigger interaction services or be orchestrated by them. Hence, both service types have in common that they realize business-specific functionality. However, interaction services are typically different from function services in the way they are consumed. As they control an interaction, they are stateful and need typically a closer integration with the respective service consumer.

In contrast to interaction services and function services, an infrastructure service is *"a software service that does not realize business functionality but technical aspects of a software system"*. Infrastructure services are often not defined and accessible in

the same way as function services with a dedicated interface providing a well-defined functionality. Rather, they often need a very specific integration into the overall technical platform and come with their own ways of interaction. Examples of infrastructure services are process orchestration, data management, data transformation, etc.

2.2 Business Service Alignment

As both business services and software services have a strong impact on the design of modern enterprise architectures, the explanation of how services are related with business process elements is a central purpose of our conceptual model. However, for understanding how the service concept fits to business processes, formalizing the general nature of business processes is a prerequisite.

According to several definitions, we define a business process as "*a self-contained, ordered sequence of business activities initiated by a defined business event (trigger), with defined input and output, a defined start state, as well as a persistent final state of value*". Hence, the purpose of each business process is the realization of a business service through the execution of one or more business activities (see Figure 2).

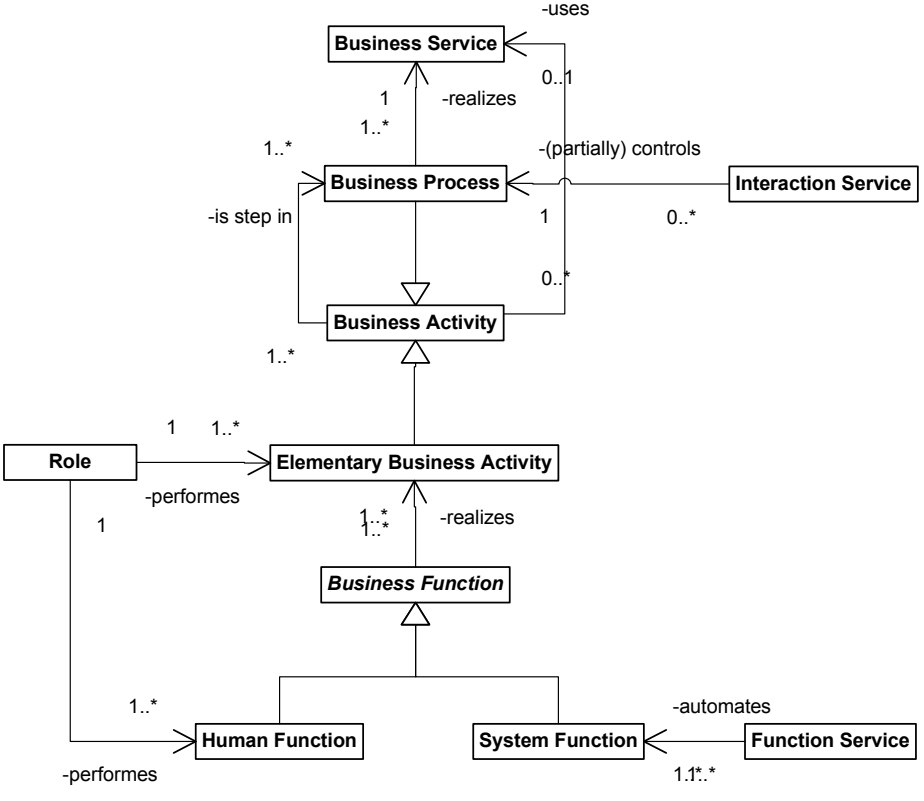

Fig. 2. Service-Business Alignment

A business activity is a step within a business process. However, business activities that are not executed by the organizational unit that provides the process' business service itself will be consumed as business services provided by other organizational units. In the transportation example, for instance, several steps within the transport chain can be outsourced to sub contractors. Hence, these sub contractors provide business services that are used for the realization of business activities in the entire transportation process.

In general, each business activity is either again a business process or an elementary business activity. Thus, each business process consists of at least one elementary business activity. An elementary business activity is *"an atomic step within a business process that ends in a persistent state"*. While "atomic" means that an elementary business unit cannot be further segmented, "persistent" indicates that each elementary business activity has to contribute a persistent value to the business process it belongs to. This means that another business activity is necessary for revocation in case of need. Furthermore, at most one person (acting as exactly one role) is allowed to perform an elementary business activity by interacting with at most one software system.

A business function "*is an atomic step within an elementary business activity*". Thereby, a business function is always executed or performed without any external interruption. Thus, each function is either performed by a human or by system why business functions are classified into human functions and system functions.

A human function is *"a business function that is performed by exactly one person in a specific role as reaction to an external trigger"*. An external trigger may be, for instance, a request from another person, a request from a system, or an environmental context change. In contrast, a system function is *"a business function that is automated by the system as reaction on an external trigger"*. An external trigger in this case may be, for instance, an explicit user request, a call from another external system, or an environmental context change that is detected via sensors.

In the context of a service-oriented application landscape, system functions are provided by function services. In contrast, the purpose of interaction services is mainly the (partial) control or execution of (long-lasting) business processes in which a multitude of humans and function services may be involved.

Services have therefore an impact on business processes on different places. Business services are realized by business processes but may also be consumed by business processes to realize business activities. Software services support the execution of business processes through the provision of corresponding control functionality or the automation of business functions within business activities. However, business services and software services are not only connected in this indirect way. Rather, they can be directly aligned. The next sub section therefore deals with the usage of software services as a channel via which business services can be requested and delivered.

2.3 Service Channels

A central idea behind service orientation has been the possibility to provide and consume business services via function services that are deployed somewhere in the internet. According to this notion, flexible and virtual business collaborations should be enabled. Thus, besides the automation of business functions, function services can also be used to invoke a business service.

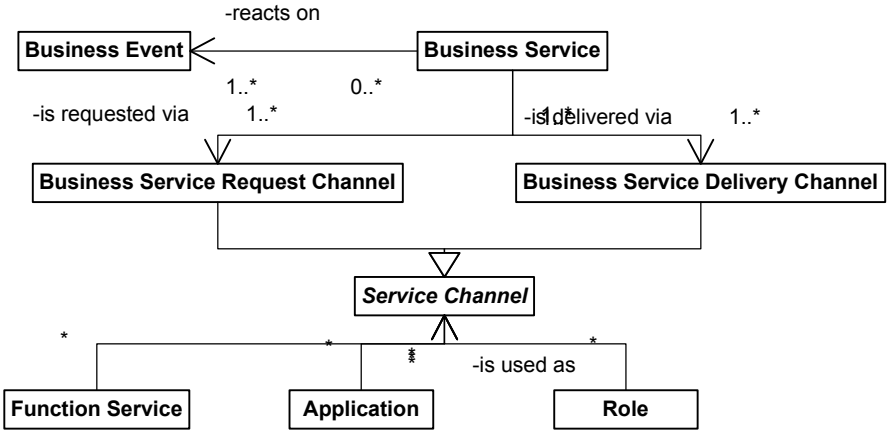

Fig. 3. Channels for Service Request and Service Delivery

In our conceptual model, we have therefore clarified the different channels through which business services can be requested and delivered (see Figure 3). In this regard, a business service delivery channel is *"a channel via which a business service is delivered to the consumer"*. Accordingly, a business service request channel is "*is a channel via which a business service is requested by perceiving a business event via the same channel*". Both the request and the delivery of a business service can be done through different channels. For instance, a (web) application can be used to request / order a product (the receipt of a corresponding order is an example for a business event) while a (human) role is responsible to deliver the ordered product to the customer. In the vision of business process collaborations, however, function services are assumed to be the main request channel, because they allow automating the request of external business services. In particular, business processes on customer side are expected to automatically request business services on provider side for realizing certain business activities through a function service's interface. If the resulting business service is then delivered electronically (via application or via function service) or via a (human) role depends on the type of business service, of course. For instance, real world business services such as transporting goods will be delivered by roles, while stock information will be delivered by function services.

In our model, it is therefore important to clearly distinguish business services and software services even if both may (especially in the future) be strongly aligned. The realization of a business services will always remain a business process, even if software services may be used to request, deliver, or even realize the business service. Nevertheless, the concept of channels allows making explicitly clear how software services and business services are related. Thus, many misconceptions on SOA can be avoided when taking this clarification into consideration.

2.4 Service Provider and Service Consumer

The traditional roles in SOA distinguish service provider, service consumer and service broker. While service provider and service consumer fit to all types of services

introduce in our conceptual model above, service broker are rather relevant for software services only.

In the context of business services, we therefore propose distinguishing the business roles of customer, solution provider, and supplier (see Figure 4). Basically, a business *role "is a set of responsibilities, rights, duties and tasks that can be taken by an organizational unit"*.

A customer is a business role that only consumes business services. Customers never provide a business service within the overall scope of interest (respectively the area of business to be managed). Thus, customers are always the final consumers of business services. This means that a customer is at the upper end of the value chain and all business services provided by the customer are out of scope for the business to be modeled. For example, when modeling the business of a logistic enterprise, the business services of the logistic enterprise's customers (e.g., an automotive manufacturer) would be out of scope.

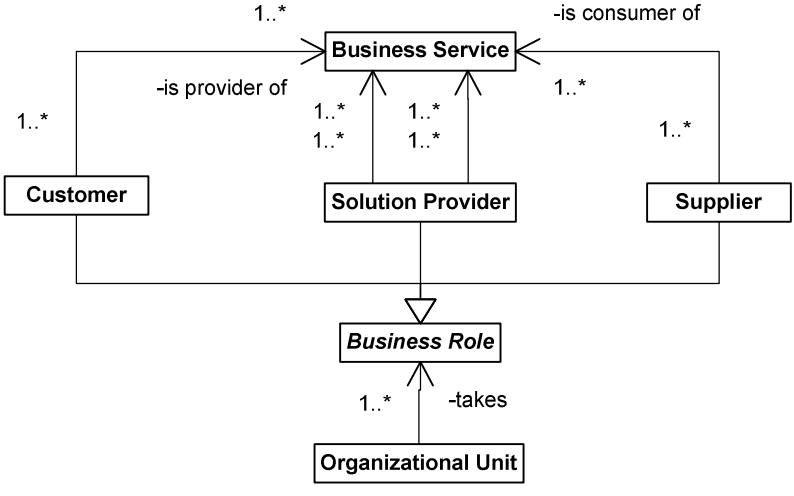

Fig. 4. Business Roles with regard to a certain Business Service

A solution provider, in contrast, is a business role that consumes and provides business services within the overall scope of interest. However, the solution provider only needs the consumed services for providing his own services and not for his own sake. In our transportation example (see section 3), the organization that manages the entire transport from a location A to a location B takes the role of a solution provider, even if transportation services of sub contractors are consumed for some sections in the transport chain.

Finally, a supplier is a business role that only provides business services and never consumes business services within the overall scope of interest. Thus, suppliers are always only providers of business services. This means that a supplier is at the lower end of the value chain in scope and the business services consumed by him are out of scope for the business to be modeled. When modeling the business of a logistics

enterprise, as mentioned above, the business services consumed by the logistics enterprise's sub contractor (e.g., a medium-sized shipping company) would be out of scope.

Through this conceptualization, the value chains that are supported by software services can be expressed in a holistic and uniform manner. Our aim to better integrate business and software engineering in the context of service-oriented enterprises can thereby be supported.

3 Example

To illustrate the applicability of our conceptual model, the previously given examples are briefly summarized, integrated and enhanced into a common example in this section. In Figure 5, the elements used in the example and their relationships are graphically shown. Business services are highlighted through light-gray boxes, while software services are highlighted through dark-gray boxes. The boxes with the dotted background represent elements of the business process hierarchy. The different roles an organization has are expressed through the lines. Solid lines depict organizational units acting as a solution provider. Dashed lines depict organizational units acting as a supplier. Finally, dotted lines represent organizational units that act as a customer within the overall scope of interest. However, it has to be noted that a suitable representation when instantiating our conceptual model is still future work. So far, we have just instantiated the conceptual model in terms of UML objects in order to provide a proof-of-concepts. We are aware that this is not the best way to represent the concepts when modeling real world settings.

Example:
The fictive logistics enterprise LOGISTICS provides a business service "All inclusive Air Freight" in which goods are transported from one location in Europe to another location on another continent (see Figure 5).

For realizing this service, LOGISTICS carries out a complex business process "Air Freight Transportation Process" that consists of several business activities. As LOGISTCS does neither have own airplanes nor own personal at each place on earth, the business activities "Delivery" and "Flight" are subcontracted to plain business service suppliers such as local shipment companies or cargo airlines. Hence, LOGISTICS consumes business service provided by third parties in order to realize its value-added business service mentioned above. LOGISTICS therefore acts as a solution provider with regard to this service.

The business service customers of LOGISTICS, which consumes the "All inclusive Air Freight" service, include companies from the automotive domain, which have to deliver spare parts to any place on earth. For ordering a transport, the customers can request the "All inclusive Air Freight" business services via a function service that is automatically invoked by the customer's ERP system. When the business event "Order received" is perceived via this function service, a new instance of the business process "Air Freight Transportation Process" is created at LOGISTICS. In particular, the interaction service "Transportation Control" within LOGISTICS' internal transport management system (TMS) is started in order to control the execution of this process instance. This interaction service then orchestrates also function services that

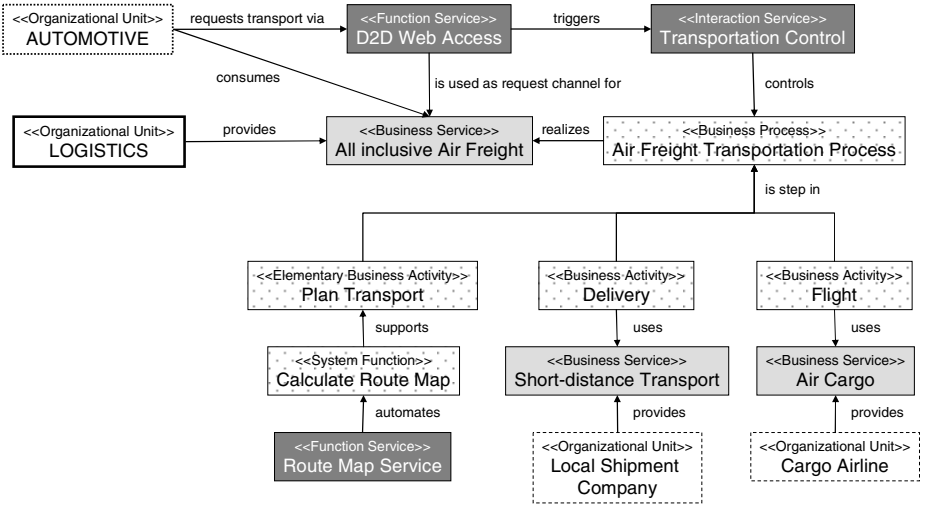

Fig. 5. Visualization of Scenario

act as request channels for the business services provided by the suppliers. Thus, the service-oriented nature of the business process is immediately reflected in a service-oriented IT landscape.

For all (elementary) business activities, which are performed by LOGISTICS itself, appropriate system support exists. As LOGISTICS aims at keeping a high flexibility in their business processes, the own IT landscape is also organized in a service-oriented way. Therefore, each elementary business activity uses system functions provided by (internal) function services. The calculation of a transportation route map is an example for such a system function that is used during the business activity "Plan Transport".

This simple example provides a first insight in the power of our conceptual model. It shows that is quite easy to explain the impact of service orientation in an integrated manner, both from a technical and from a business point of view. In particular, an understanding of the dependencies between different organizational units based on their business service exchanges allows a systematic identifying of appropriate software services support. Hence, representing services in business processes supports assessing the benefits of service-oriented technology in this regard.

4 Conclusion and Outlook

Even if SOA has received much attention, there is still no common definition of what a SOA is or what a SOA should provide. Rather, many different definitions can be currently found in literature. While all these different definitions and views have their eligibility, they tend to neglect that their might be a more holistic view on SOA.

In this paper, we have therefore introduced a consolidated picture on service orientation that is aligned with traditional business process concepts in order to explain

how business issues, application landscapes and technical details are related and how one aspect can influence another one. The views shown in this paper have represented a sub set of our entire conceptual model that also includes many other aspects of SOA and also explicit links to traditional software engineering artifacts.

The purpose of the views presented here is to explain which impacts the service-oriented paradigm may have on business processes as well as on the development of supporting software systems. We consider this consolidation as an essential step for establishing methods for a better business IT alignment as well as more systematic and integrated business and software engineering in the context of service-oriented enterprises.

Currently, we finalize other views of our conceptual model. Furthermore, we have recently started to develop integrated software and business engineering methods that address the systematic alignment of business and IT in service-oriented setting according to our conceptual model. Finally, we are investigating how an instantiation of our conceptual model should be represented.

Acknowledgement

The results of this paper have been developed in the context of the ADiWa project funded by the German Federal Ministry of Education and Research, grant no. 01IA08006F.

References

1. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice-Hall, Englewood Cliffs (2004)
2. Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R.: Service-Oriented Architecture Compass. IBM developerWorks (2005)
3. Erl, T.: Service-Oriented Architecture – Concepts, Technology, and Design. Prentice-Hall, Englewood Cliffs (2005)
4. Masak, D.: SOA? Serviceorientierung in Business und Software. Springer, Heidelberg (2007)
5. OASIS: Reference Model for Service-Oriented Architecture 1.0. Committee Specification 1 (2006), http://www.oasis-open.org/committees/download.php/19361/soa-rm-cs.pdf
6. Dostal, W., Jeckle, M., Melzer, I., Zengler, B.: Service-orientierte Architekturen mit Web Services. Elsevier, Amsterdam (2005)
7. OASIS, SOA Reference Architecture, http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf
8. Engels, G., et al.: Quasar Enterprise. dpunkt.verlag (2008)