View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

From Cross-Organizational Business Process to Public Services

Martin Nussbaumer
Karlsruhe Institute of Technology, martin.nussbaumer@kit.edu

Tobias Vogel
University of St. Gallen (Switzerland), Tobias.Vogel@unisg.ch

Andrea Fuchsloch
SAP Research St. Gallen (Switzerland), andrea.fuchsloch@sap.com

Follow this and additional works at: http://aisel.aisnet.org/amcis2010

Recommended Citation
Nussbaumer, Martin; Vogel, Tobias; and Fuchsloch, Andrea, "From Cross-Organizational Business Process to Public Services" (2010). *AMCIS 2010 Proceedings*. 268.
http://aisel.aisnet.org/amcis2010/268

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

From Cross-Organizational Business Process to Public Services

Andrea Fuchsloch
SAP Research St. Gallen (Switzerland)
andrea.fuchsloch@sap.com

Alexander Schmidt
University of St. Gallen (Switzerland)
alexander.schmidt@unisg.ch

Tobias Vogel
University of St. Gallen (Switzerland)
tobias.vogel@unisg.ch

Martin Nussbaumer
Karlsruhe Institute of Technology
martin.nussbaumer@kit.edu

ABSTRACT

Service-oriented architectures promise flexible process integration in heterogeneous environments, particularly in cross organizational contexts. Therefore, a systematic approach for deriving service definitions from cross-organizational business processes is required. The paper at hand presents a structured, model driven approach that allows for cross-organizational integration with service-oriented concepts and technologies based on a cross-organizational business process. The resulting method is specified by means of a metamodel and a procedure model. Taking into account existing approaches in the research field, the paper focuses on the activities for the service design specification and the subsequent implementation of the public services. Feasibility of the method is shown by applying it in a Business-to-Government (B2G) real-world scenario, namely the collaboration between a consigning company and customs authorities according to the new European harmonized customs procedures.

Keywords

Cross-organizational business processes, service-oriented architecture, service design, B2G integration.

INTRODUCTION AND MOTIVATION

With increased levels of external process integration and a growing number of electronic interaction between enterprises and governmental authorities there is a demand to plan process and system interfaces in the Business-to-Government (B2G) domain systematically and to implement them electronically. Some scholars favour service orientation techniques and web service standards for cross-organizational process integration, e.g. (Daniel and White, 2005), (Gosain, 2007, p. 64-65), (Goethals et al., 2005, pp. 8ff.), (Ratnasingam, 2004) and (Vidgen et al., 2004). As autonomy, heterogeneity and dynamism in technical and business terms rises at the borders of organizations (Singh and Huhns, 2005, pp. 7-10), it is envisioned that in the future services could be leveraged to exchange electronic business documents for ensuring better interoperability and enabling a seamless and extensively automated coordination and communication in the B2G domain (Daniel et al., 2004).

At present there are no well established methods which realize cross-organizational integration with service-oriented concepts and technologies on the basis of a commonly defined or standardized cross-organizational business process (cp. Colombo et al., 2008; Legner and Wende, 2007). Existing approaches focus either on the requirements analysis and modeling of business processes virtually neglecting the underlying implementation specifics (e.g. (Bresciani et al., 2004) or (Desai et al., 2009)) or they do not sufficiently consider cross-organizational business process models during service design (e.g. (Baghdadi, 2005) or (Erl, 2005)). The objective of our approach is to close this gap and develop a method which derives service specifications from cross-organizational processes systematically incorporating business requirements in the specification of service interfaces and operations.

The remainder of the paper is structured as follows: In the background section we specify the research method we have applied and set forth the conceptual foundations our approach is based on. The third section presents related work and identifies the need for further research. Based on the analysis of related work we derive requirements that are relevant for systematic approach satisfying our research goal. Thereafter, our method is presented by means of a metamodel and a procedure model. In the penultimate section we evaluate our method by applying it in an eCustoms scenario, in which a

company and customs authorities interact according to the new European eCustom declarations procedures (cp. European Commission, 2007). The final section provides a short summary of the insights which we have gained from our research and reveals the impact on further research.

BACKGROUND

Research Approach

For the development of our method we follow the Design Science Research (DSR) approach (cp. Hevner et al., 2004; March and Smith, 1995) The design science approach focuses on the development of effective solutions for practical problems (Hevner et al., 2004, p. 81). Consequently, DSR projects are usually initiated by a problem found in an actual application environment. In the desire of solving the given problem the researcher draws on the knowledge and experience found in the respective application domain (section 3) (Hevner, 2007, pp. 88-89). The core of DSR consists in creating the solution, in our case a method for deriving public service definitions from cross-organizational processes. The design activity represents an iterative process that starts with gathering requirements, continues with constructing the method, and then proceeds with evaluating the method through application in a real-world scenario to derive necessary adjustments (section 5) (Rossi and Sein, 2003).

For ensuring a stringent construction of our method, we used Method Engineering as a framework for defining our artifact (cp. Brinkkemper, 1996; Gutzwiller, 1994). Method Engineering denotes the systematic approach of designing, constructing and adapting methods for the development of information systems (Brinkkemper, 1996, p. 276). As its two main constituents, a method embodies a metamodel, i.e. a conceptual data model of the design elements of the method, as well as a procedure model defining the chronological order of design activities and, thus, the process to guide achievement of the intended results (Karlsson and Agerfalk, 2004, p. 621; Leist and Zellner, 2006, p. 1548). The paper at hand presents both components for our method.

Service Orientation

A service-oriented architecture (SOA) is a multi-layer distributed IS architecture, which encapsulates application data and functionality behind a well-defined, stable interface in a technological neutral manner to foster reuse and to be orchestrated in workflows (Alonso et al., 2003, p. 8; Arsanjani, 2004; Papazoglou, 2008, p. 29). Services are the core building blocks of a SOA, which can be realized by different technologies such as Web Services (Web Services Architecture) relying on the W3C standards (SOAP, WSDL) or RESTful services (Umapathy and Purao, 2007), which follow the principles of the Representational State Transfer paradigm as defined by(Fielding, 2000). In the context of machine-to-machine integration between organizations we prefer Web Services following the argumentation of (Pautasso et al., 2008, p. 803), who propose to use RESTful services for tactical, ad hoc integration over the Web (à la Mashup) and to prefer WS-* Web services in professional enterprise application integration scenarios with a longer lifespan and advanced QoS requirements.

According to (Motahari Nezhad et al., 2007), a service specification is considered as a triplet P = (D; M; O), where D is the set of data types of the service, M is the set of messages or business documents exchanged as part of operation invocations, and O is the set of operations supported by the service. The design of service specification is considered to be a critical point. Based on a broad literature review, Legner and Heutschi identify four fundamental principles of service design (Legner and Heutschi, 2007, pp. 1645-1647): (1) separation of interface logic and implementation logic, (2) interoperability through technical and business standards, (3) loose coupling, and (4) process oriented service identification. Especially in the crossorganizational context, interoperability and process orientation are regarded as essential (Legner and Vogel, 2008, p. 50).

RELATED WORK

The implementation of SOA for cross-organizational business process integration requires effective methods, which support the systematic and structured design and implementation of services (Baghdadi, 2005; Erl, 2005; Feuerlicht and Meesathit, 2004). At present there are no well established methods or design principles continuously supporting a model based derivation of service definitions based on the requirements of cross-organizational integration (Colombo et al., 2008, 377). The development of appropriate methods is subject to current research as well as to this paper.

As starting point of our research we have investigated existing approaches, which focus on specific aspects of service design in more detail. For lack of space, we restrict presentation of the approaches in this paper to a brief evaluation of the four most relevant of them focusing on their shortcomings for our research goals: (1) Baghdadi`s Business Model for Deploying Web Services approach (Baghdadi, 2005) corresponds to a pure bottom-up procedure. Based on the data and the knowledge of a

business he uses factual dependencies for the derivation of services. The method largely omits the criteria of process orientation and results in fine-grained services. (2) In the Design Method for Interoperable Web Service (Feuerlicht and Meesathit, 2004) suggest to use data normalization techniques for the identification of services. The identified services are a result of decomposition and combination of elementary activities. However, Feuerlicht does not mention concrete criteria to achieve appropriate service granularity and lacks a specification of the interface as well. (3) (Colombo et al., 2008) propose a Methodology for E-Service Design that includes a comprehensive procedure model consisting of the three phases, namely requirements specification, conceptual design and implementation. The approach puts a strong focus on requirements analysis and describes the actual service design and implementation very scarcely, omitting a detailed discussion of the underlying principles for service design. Moreover, the BPEL-based service orchestration favors a central intermediary who is responsible for implementing and maintaining the BPEL service. (4) In the Web Service Development Lifecycle (Papazoglou, 2008) introduces a method, which encompasses all phases of an SOA implementation using web services. However, the model based derivation of service interfaces is only addressed superficially. Design results and concrete criteria for service cut are not clearly specified.

REQUIREMENTS

Based on the stated problem we have derived five categories of requirements, which the method should meet. The category process orientation focuses on the support of cross-organizational business processes and thus guaranteeing Business/IT alignment. In terms of service design we derive the requirements interface design, interoperability and cohesion & coupling. The category service portfolio design addresses an adequate service granularity of the derived services. For each category the following table lists the criteria which the fulfillment of the requirements is determined against.

Requirements		**Description**
Process Orient-ation	Assistance of cross-organizational processes	The method should support cross-organizational process, enabling business process integration across company borders.
	Top-down approach	Services must be derived based on the goal of the business process they support.
Interface Design	Contract first approach	The interface design, i.e. the specification of the details of the implementation, should be carried out before the service implementation.
	Full specification of service interfaces	The interface specification should cover all aspects required for the usage of the service.
Inter-operability	Technical standards	Open, platform-independent technical standards should be used, e.g. SOAP, WSDL, XSD.
	Semantic standards	Open and established semantic standards should be used, e.g. RosettaNet.
Cohesion and Coupling	High level of service cohesion	The data and functions of a service should be highly interdependent.
	Loose coupling	Services should implement an asynchronous and document based communication style.
Service Portfolio Design	Adequate service granularity	The scope of functionality a single service provides should lead in a balanced granularity according to the cross-organizational business process.
	Criteria for service cut	Well defined criteria should support the service cut.

Table 1. Requirements for the evaluation of service design approaches

Although the existing approaches presented in the previous section provide valuable concepts for our research and are thus considered in our work, they do not cover the entire stack of requirements. This motivates our research goal to develop a comprehensive method that closes this gap and satisfies the derived requirements completely.

METHOD FOR DERIVING PUBLIC SERVICE DEFINITIONS FROM CROSS-ORGANIZATIONAL BUSINESS PROCESSES

In order to address the requirements presented in the previous chapter, we propose a method for the model driven derivation of services based on cross-organizational business processes. In the following sections, we start off by presenting the metamodel of our method before the proposed procedure model is introduced in its generic form. Subsequently, we describe the main phases and activities of the procedure model in detail.

Metamodel

The metamodel provides an overview of the design elements used within our method and their relationships to each other. In contrast to (Colombo et al., 2005) who propose a generic conceptual model for describing service-oriented system, but mainly focus on the main actors and their interactions, our metamodel considers the correlations between cross organizational business processes (Business Process View), their representation in service-based information systems (Information System View) and their platform specific representation (see Figure 1). The dark gray elements on the bottom of the figure belong to the WS-* Web Service View that shows a platform specific representation of information system elements as applied within our approach.

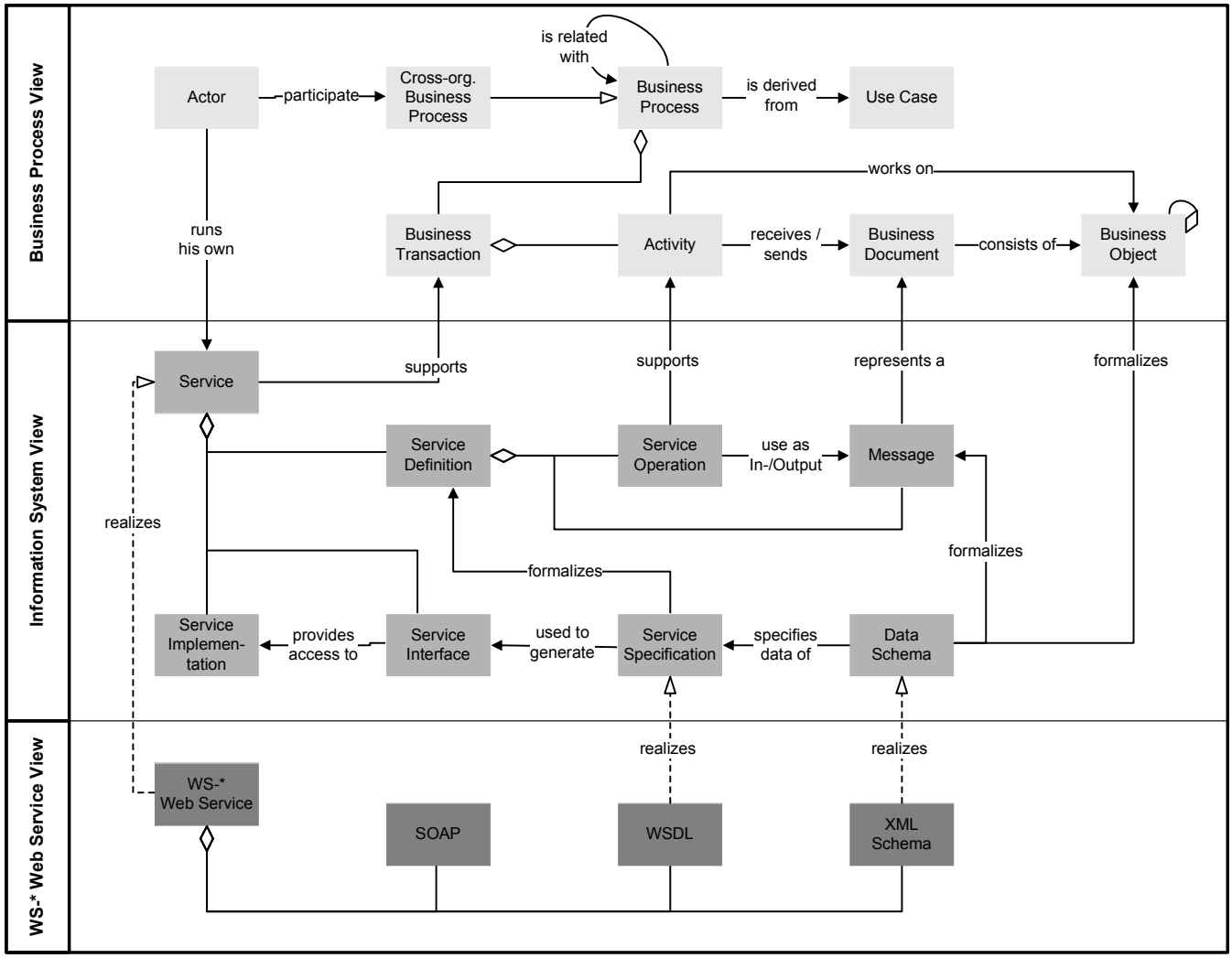

Figure 1. Metamodel of the proposed Method

The core element of the Business Process View is the cross-organizational business process, which involves more than one organization (Legner and Wende, 2007, p. 1645). An actor, e.g. an organization, participates in cross-organizational business processes and runs its own information systems. A business process is composed of several business transactions with each transaction consisting of a number of activities. An activity is able to send and receive business documents, which are composed of business objects.

The top level element of the Information System View is a service. A service should at least provide the functionality that is needed to support a complete business transaction. The service definition, which is part of the service, consists of service operations and messages. Service operations use messages as in- and outputs. Messages, as well as business objects, are formalized in a data schema. The data schema is part of the service specification that formalizes the service definition. The service specification is used to generate the implementation of the service interface.

The WS-* Web Service View is a technology specific view that specifies how the information system elements and services are implemented by using Web Services technology (see section Service Orientation). SOAP, WSDL and XML schema are the core standards for web services and are completely supported by SOA platforms. XML Schema is used to realize the data schema and formalizes the business objects and messages as xsd:complex types. The XML schemas are imported into the WSDL specification, where the service operations and the exchanged messages are specified.

Procedure Model

The purpose of the developed procedure model is to facilitate the structured, model driven derivation of services based on cross-organizational business processes. Based on the Rational Unified Process (IBM, 2007), the procedure model shown in Figure 2 consists of three phases, namely Analysis, Design and Implementation, and comprises a total of twelve activities. While most of the existing approaches mainly focus on the Analysis phases (cp. Bresciani et al., 2004; Colombo et al., 2008; Papazoglou, 2008), our procedure model focuses on the derivation of service definitions as well as the subsequent implementation of the public services. The procedure follows an iterative approach in a sense that after starting with the analysis of a single public business process the scope of the method can be widened incrementally during the Design phase by analyzing further processes. The activities of each phase will be explained in more detail in the following three sub-chapters.

Analysis

The Analysis phase starts off by modeling the first cross-organizational business process. We based this initial phase of our procedure model on UN/CEFACT's Modeling Methodology (UN/CEFACT, 2009). The first two activities cover the process view und result in UML use case diagrams (*Activity 1: Modeling of Use Cases*) and UML activity diagrams (*Activity 2: Modeling of Activities*). The information view of a cross-organizational business process including the exchanged messages is modeled within *Activity 3: Modeling of Message Exchange* and results in a UML sequence diagram, in which the identified actors are assigned to object instances.

Finally, *Activity 4: Identification of Business Objects* is used for the detection of business objects which the exchanged messages are composed of. Business objects are identified based on the Document Engineering approach of Glushko and McGrath (cp. Glushko and McGrath, 2005). More precisely, we use step four and five of their eight step approach, since we already analyzed the business process in the previous activities. The first step is to build a document repository, which includes all exchanged messages we have identified in Activity 3, as well as additional meta-information. The next step is to analyze each document by breaking it down into information and structure components. After all documents have been analyzed, the identified components have to be consolidated in a single table, which is the design result of Activity 4.

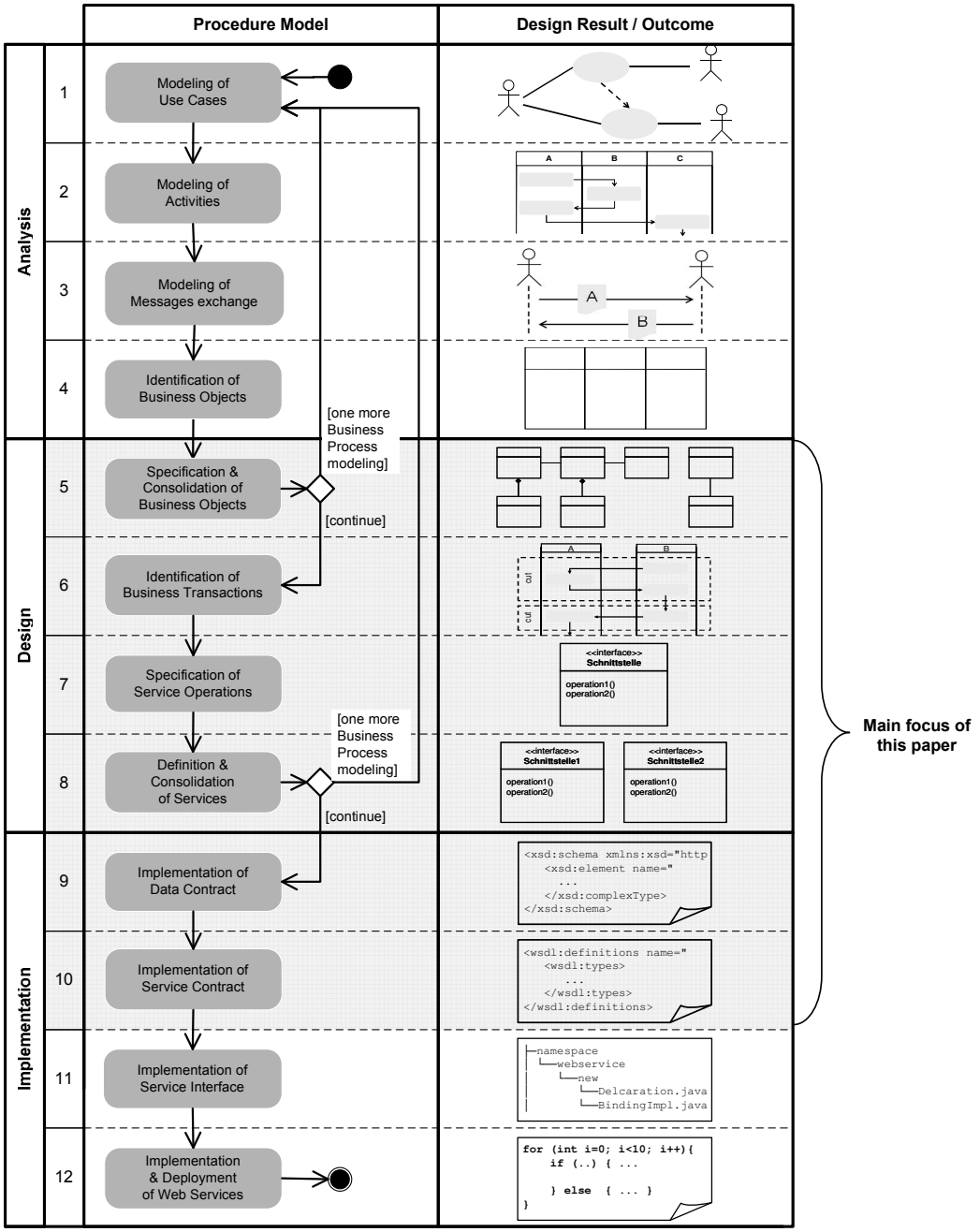

Figure 2. Procedure model - From Cross-organizational Business Process to Public Services

Design

The Design phase starts with *Activity 5: Specification & Consolidation of Business Objects*. Based on the consolidated table of Activity 4, the structure components are mapped to business objects while the information components serve as attributes of the business objects. The outcome of Activity 5 is a UML class diagram that illustrates the specified business objects as classes with corresponding attributes. In doing so, we obtain a consolidated data model for the whole business process.

The next step is *Activity 6: Identification of Business Transactions*. According to (OASIS, 2002, p. 17) a business transaction can be defined as a consistent change in the state of a business relationship between two or more actors. Therefore, a business transaction can be understood as a sequence of activities, which forms a logical unit. The principle of atomicity, meaning that a business transaction should either be executed completely or not at all, applies to business transactions as well. Figure 3

illustrates schematically the relationship between business transactions and the control flow of a cross-organizational business process. The identified business transactions serve as input for the following activity

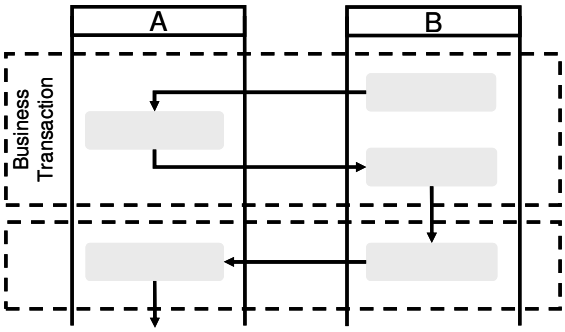

Figure 3. Identification of Business Transactions

Within *Activity 7: Specification of Service Operations* the service operations are derived by using the mapping table shown in Table 2. To derive the required service operations per actor, the first column of the mapping table covers actors a_i to a_n of set N. The second column lists all business transactions t_j to t_m per actor, in which he participates. The next step is to identify all activities that are essential for cross-organizational collaboration, i.e. that either send or receive business documents. Hence, the third column depicts the collaboration activities z_{kij} to z_{pij} for each actor a_i and each transaction t_j. After the collaboration activities have been identified, they need to be classified. An identified activity could either invoke a service operation of another actor, receive a service call, or reply to a service call. Activities in the receive class are able to provide a synchronic, an asynchronous, or no reply at all.

Actors a_i - a_n	Transaction t_j – t_m	Activity z_{kij} - z_{pij}	Activity Type	Invoked Operation	Provided Operation	Operation Type
a_1	t_1	z_{111}	<invoke>			<one-way>
		…	<receive>			<request-response>
		z_{p11}	<reply>			<notification>
	…	…	…	…	…	…
	t_m	z_{11m}	<invoke>			<one-way>
		…	<receive>			<request-response>
		z_{p1m}	<reply>			<notification>
…	…	…	…	…	…	…
a_n	t_1	z_{1n1}	<invoke>			<one-way>
		…	<receive>			<request-response>
		z_{pn1}	<reply>			<notification>
	…	…	…	…	…	…
	t_m	z_{1nm}	<invoke>			<one-way>
		…	<receive>			<request-response>
		z_{pnm}	<reply>			<notification>

Table 1. Mapping table for the derivation of service operations

Based on the classified activities the service operations can be derived. Activities of the invoke class call service operations of other actors. These service operations are assigned in the fifth column (Invoked Operation). Thereafter, the operation types need to be defined. In WSDL there are four possible operation types: one-way, request-response, solicit-response and notification. If the invoked operation is expected to provide a response, it has to be clarified whether the response is going to be synchronic or asynchronous. If a synchronic response is expected, the operation has to be typed as request-response and no other operations have to be defined. Otherwise an additional service operation has to be defined to receive the response. In that case, both operations are one-way (see Firgure 4).

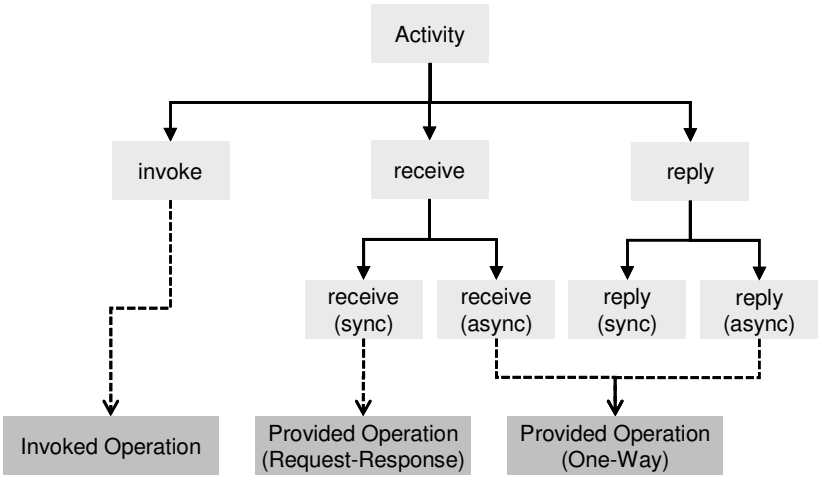

Figure 4. Derivation of service operations based on classified activities

The service operations, which have to be provided by each actor are listed in column six (Provided Operation) of the mapping table. The last step of Activity 7 is to add the input and output messages of the operations. At this point a meaningful name has to be assigned to each message.

The identified service operations are aggregated to services in *Activity 8: Definition & Consolidation of Services*. To obtain a suitable service granularity, it is essential to investigate additional criteria concerning the service design. The first criteria for service cut are the actors, requiring the maximum granularity of a service to comprise all service operations of an actor. Second, one service must at least support one complete business transaction, which means that service operations belonging to the same business transactions need to be implemented by the same service. Therefore, the minimal set of service operations consists of those operations that belong to a particular business transaction. Depending on the complexity of the business processes logical service cohesion (cp. Feuerlicht and Meesathit, 2004; Papazoglou, 2008) should be considered as third criteria. Consequently, service operations, which are mutually independent but logically related, should be implemented by the same service. Nevertheless, indivisibility of a business transaction still has to be considered.

Implementation

The Implementation phase starts with *Activity 9: Implementation of Data Contract* that formalizes the specified business objects and business documents. The implementation of the data contract is based on the XML Naming and Design Rules (XML NDR) of UN/CEFACT (UN/CEFACT, 2006). Accordingly, all business objects are specified in one single XML schema using the namespace *xmlns:bom*. Each business object is mapped to a *complexType* element while the attributes are represented as elements within the *complexType*. For each message, being composed of several business objects as specified in the XML namespace *xmlns:bom*, a separate XML schema is defined. The implementation of the data contract can be automated by software tools for the most part.

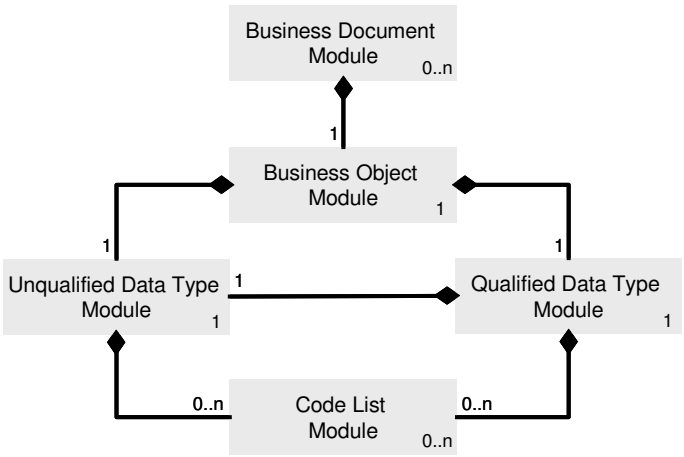

Figure 4. Model of modularity based on UN/CEFACT

Activity 10: Implementation of Service Contract is supported by several graphical software tools. Therefore, there is no need to type the WSDL syntax manually. By starting with the abstract part of the WSDL file, the first step is to import the already formalized messages, into the *wsdl:types* element. The service operations specified in Activity 8 are mapped to *wsdl:operation* elements and grouped in the element *wsdl:portType*. Based on the operation types *wsdl:input* and *wsdl:output* elements are assigned to the operations, which refer to *wsdl:message* elements. As binding protocol *SOAP* has to be defined and as *soap:binding* style *document* has to be specified.

The resulting WSDL files are used within *Activity 11: Implementation of Service Interface*. The implementation of the service interface comprises two aspects: the generation of skeletons and the transformation of the specified data types into the desired programming language. Both aspects can be automated by software tools. Based on the WSDL file and the referred XML schema files the skeletons can be generated automatically.

Activity 12: Implementation & Deployment of Web Services comprises the implementation of the business logic in the desired programming language and development environment as well as the deployment of the developed service source code on an application server resulting in executable web services.

PROOF OF CONCEPT AND EVALUATION

Proof of Concept

The proposed procedure model has been applied in a real-world scenario. The scenario originates from one of the Living Labs of the EU-funded research project ITAIDE[1], which aims at developing electronic customs solutions. With a total of approximately 175 million paper-based customs declarations filed in the European Union per year (Kuhnen, 2009, p. 3), customs process exhibit a significant potential for improvement in efficiency. Within the Multi-Annual Strategic Plan (MASP) the European Commission announced the transformation of the export process into a paperless electronic-based process to realize a more efficient customs clearance (cp. European Commission, 2007).

Analysis

Starting with Activity 1, we identified four use cases for the export process of the European Union in which four actors participate: the consignor, the office of departure (OOD), the office of exit (OOE) and the federal office of statistics. The first use case is named export declaration and is initiated by the consignor, who sends his export declaration to the office of departure. In context of Activity 2 the already identified actors are assigned to the swim lanes of the UML activity diagram. The export procedure starts off by sending an export declaration to the office of departure. The OOD receives and reviews the declaration. If not all needed information is included the OOD sends a refusal of acceptance to the consignor. If the declaration is accepted, a unique Movement Reference Number (MRN) is assigned and the acceptance including the MRN is send to the consignor. The identified actors are assigned to object instances of a UML sequence diagram within Activity 3. At best, the export process comprises 14 exchanged messages. Activity 4 starts with the document repository. For the export

[1] For further information on the project see http://www.itaide.org

process the document repository consists of the Single Administrative Document (SAD), which is the documentary basis for customs declarations in the EU, the web pages of the European Commission and corresponding laws. The consignor fields of the SAD form needs to be filled out with the name and the address of the consignor. Entries like *name*, *street* and *city* are information components, while *consignor* and *address* are structure components. Table 3 shows an extract of the resulting, consolidated information and structure components.

Component Name	Business Rule	Cardin-ality	Data Type	Type	Object Name
Consignor		1		Structure	Consignor
Last Name	Has to be filled in, if no firm name is given	0..1	Name	Information	
First Name	Has to be filled in, if no firm name is given	0..1	Name	Information	
Firm	Has to be filled in, if no first & last name is given	0..1	Name	Information	
Adress		1		Structure	Address
Street		0..1	Text	Information	
House Number		0..1	Text	Information	
Postal Code		0..1	Text	Information	
City		0..1	Name	Information	
Country	Country Code	0..1	Country Identifier	Information	

Table 3. Information and structure components of the export process

Design

At the beginning the design phase, within *Activity 5: Specification & Consolidation of Business Objects* the identified components are mapped to business objects. In terms of the export process example we have specified the business objects address and consignor. The first business transaction of the export process is named ExportDeclaration and ends with acceptance or rejection of the declaration, while the second transaction is called ReleaseGoodsForExport and starts with the presentation of the good to customs (*Activity 6: Identification of Business Transactions*). By using the mapping table of *Activity 7: Specification of Service Operations* we have identified for actor consignor and transaction ExportDeclaration three activities that are essential for cross-organizational collaboration, namely send Export Declaration, receive refusal of acceptance and receive acceptance and MRN, see table 4. Send Export Declaration is an invoke activity and we have named the invoked operation submitExportDeclaration. This operation must be provided by the Office of departure which also can be seen from the mapping table. The signature of the first operation for the Office of departure is submitExportDeclaration (ExportDeclarartion): ControlResult.

Actor a_i - a_n	Transaction t_j - t_m	Activity z_{kij} - z_{pij}	Activity Type	Invoked Operations	Provided Operations	Operation Type
Consignor	Export Declaration	send Export Declaration	invoke	submitExport Declaration		Request-Response
		receive refusal of acceptance	reply (sync)			
		receive acceptance and MRN	reply (sync)			
…						
Office of Departure	Export Declaration	receive Export Declaration	receive (sync)		submitExport Declaration	Request-Response
		send refusal of acceptance	reply (sync)			
		send acceptance and MRN	reply (sync)			

Table 4. Mapping table of the export process

The specified service operations have been grouped per actor and thus satisfy the first as well as the second criterion for the service cut. Since the maximum number of service operations per actor is four, the criterion of logical service cohesion does not apply here.

Implementation

The business objects specified in the data model (Activity 5) have been exported as XML schema using namespace bom (business object module) within Activity 9. The resulting XML schemas for business documents, represented by messages, are using the namespace cdm, which stands for customs documents module. In our example the message cdm:ExportDeclaration that represents a business document has its own XML schema. It references to the complexType *bom:DeclarationType* in the business object schema *(bom:DeclarationType)*. Activity 10 is supported by several graphical software tools, therefore we have modeled the WSDL file with a graphical editor. By starting with the abstract part of the WSDL file, the first step is to import the already formalized messages into the *wsdl:types* element. The service contract for the OOD has to import the XML schema for the business document export declaration *<xs:import namespace=…schemaLocation= "…ExportDeclaration.xsd"/>*. The service operations specified in Activity 8 are mapped to *wsdl:operation* elements (e.g. *<wsdl:operation name="submitExportDeclaration">*) and grouped in the element *wsdl:portType*. Figure 6 shows the resulting WSDL file for the OOD. Based on the WSDL file and the referred XML schema files the skeletons have been generated by using thinktecture`s WSCF (Web Services Contract First) add-on (Activity 11). For Activity 12 we used Microsoft Visual Studio and C# as programming language to implement the business logic before deploying the service on an application server.

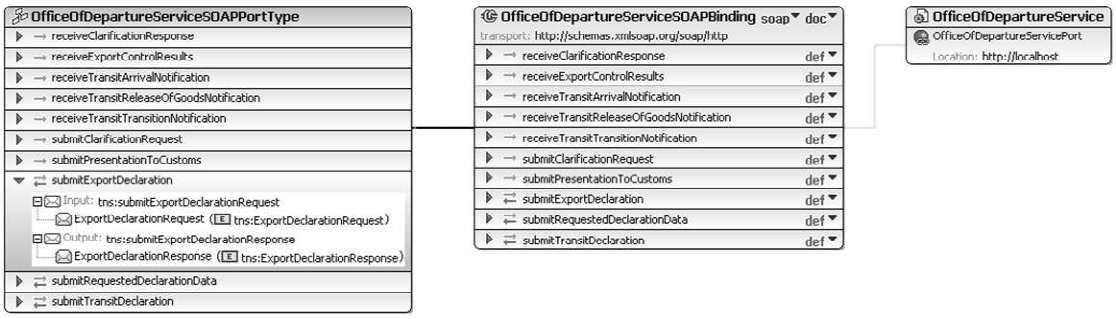

Figure 5. Modeling of the WSDL description

Evaluation

The proposed method is evaluated on the basis of the criteria derived in the requirements chapter. The following table summarizes the evaluation of the developed method and verifies the fulfillment of the defined requirements.

Requirements		**Evaluation**
Process Orient-ation	Assistance of cross-organizational processes	The methods' starting point is a cross-organizational business process. The actors and their collaboration are in the center of the action.
	Top-down approach	The derivation of the service definitions follows the top-down approach, as the whole cross-organizational business process is considered first and then incrementally decomposed. On the contrary, the identification of the business objects follows a bottom-up approach.
Interface Design	Contract first approach	The design of the service interface is a central design element of the suggested method. The specification of the service interfaces is entirely done before the implementation.
	Full specification of service interfaces	The specification of service interfaces meets all technical aspects. In addition to this, the activity diagrams of the to be supported business transactions could be considered, to also have a semantic description of the service definition.
Inter-operability	Technical standards	The method is using continuously open, technological and platform independent standards.
	Semantic standards	The suggested method uses UN/CEFACT standards including CCTS as a semantic standard for specification of the business objects.
Cohesion	High level of service cohesion	The used data and functions are highly interdependent, since they are derived from one or several related business processes.

Requirements		Evaluation
and Coupling	Loose coupling	The method supports a synchronous as well as an asynchronous, message-based communication of the services.
Service Portfolio Design	Adequate service granularity	The capacity of a service always includes at least a complete business transaction and all necessary operations of a cross-organization business process at a max. Furthermore, the service capacity can be adapted to the context of the business process through the criterion of logical cohesion.
	Criteria for service cut	Within activity seven and eight the method specifies several Criteria for service cut.

Table 5. Evaluation of different service design approaches

CONCLUSION AND OUTLOOK

Service-oriented architectures promise flexible process integration in heterogeneous environments, particularly in crossorganizational contexts. However, a systematic approach to analyzing inter-organizational business processes and transforming them into deployable services is still missing. The paper at hand, therefore presented a structured, model driven approach that allows for cross-organizational integration with service-oriented concepts and technologies based on a commonly defined public process. The particularity of the method is the systematic derivation of service specifications from cross-organizational processes. Applicability of the method was shown by applying it as a proof of concept in a B2G realworld scenario, namely the collaboration between a consigning company and customs within an export process.

Further research should focus on applying the method in further cases. This would not only increase validity of our approach but beyond that would provide insights on the potential need for refinement of the method. Moreover, we encountered considerable improvement potential concerning efficient support of our method with appropriate software tools. For the time being, the tool support is rather fragmented as we were forced to use different tools for developing the models within the different phases. An integrated tool support that increases automation of some activities (particularly during the design phase) and enables the developer to directly access design results of previous phases within the same tool would significantly speed up the development process.

REFERENCES

1. Alonso, G., Casati, F., Kuno, H. and Machiraju, V. (2003) Web Services: Concepts, Architectures and Applications, Springer, Berlin et al.
2. Arsanjani, A. (2004) Service-oriented modeling and architecture - How to identify, specify, and realize services for your SOA, http://www-106.ibm.com/developerworks/webservices/library/ws-soa-design1, 22.01.2009.
3. Baghdadi, Y. (2005) A business model for deploying Web services: A data-centric approach based on factual dependencies, *Information Systems & E-Business Management* (3:2), pp. 151-173.
4. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. (2004) Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Systems* (8:3), pp. 203-236.
5. Brinkkemper, S. (1996) Method Engineering - Engineering of Information Systems Development Methods and Tools, *Information and Software Technology* (38:4), pp. 275-280.
6. Colombo, E., Francalanci, C. and Pernici, B. (2008) Modeling Cooperation in Virtual Districts: A Methodology for e-Service Design, *International Journal of Cooperative Information Systems* (13:4), pp. 369-411.
7. Colombo, M., Di Nitto, E., Di Penta, M., Distante, D. and Zuccala, M. (2005) Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems, *Proceedings of International Conference on Service-Oriented Computing (ICSOC 05)*, Lecture Notes in Computer Science (LNCS 3826), Amsterdam, pp. 48-60.
8. Daniel, E. M. and White, A. (2005) The future of inter-organisational system linkages: findings of an international Delphi study, *European Journal of Information Systems* (14:2), pp. 188-203.
9. Daniel, E. M., White, A. and Ward, J. (2004) Exploring the role of third parties in inter-organisational web service adoption, *Journal of Enterprise Information Management* (17:5), pp. 351-360.
10. Desai, N., Chopra, A. K. and Singh, M. P. (2009) Amoeba: A Methodology for Modeling and Evolving Cross-Organizational Business Processes, *ACM Transactions on Software Engineering and Methodology* (19:2), Article 6.

11. Erl, T., (2005) Service-Oriented Architecture: Concepts, Technology, and Design Prentice Hall International, Upper Saddle River, New York, USA.
12. European Commission (2007) Electronic Customs Multi-Annual Strategic Plan (MASP) Rev. 8, DG TAXUD, Brussels, Belgium.
13. Feuerlicht, G. and Meesathit, S. (2004) Design Method for Interoperable Web Services, *The 2nd International Conference On Service Oriented Computing - ICSOC '04*, New York, USA, pp. 299-307.
14. Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures, University of California, Irvine, CA.
15. Glushko, R. J. and McGrath, T. (2005) Document Engineering - Analyzing and Designing Documents for Business Informatics and Web Services, MIT Press, Boston, MA.
16. Goethals, F., Vandenbulcke, J. and Lemahieu, W. (2005) Two Basic Types of Business-to-Business Integration, *International Journal of E-Business Research* (1:1), pp. 1-15.
17. Gosain, S. (2007) Realizing the vision for web services: Strategies for dealing with imperfect standards, *Information Systems Frontier* (9:1), pp. 53-67.
18. Gutzwiller, T. A. (1994) Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg.
19. Hevner, A. R. (2007) A Three Cycle View of Design Science Research, *Scandinavian Journal of Information Systems* (19:2), pp. 87-92.
20. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) Design Science in Information Systems Research, *MIS Quarterly* (28:1), pp. 75-105.
21. *IBM (2007)* IBM Rational Unified Process - Improving project performance with proven, adaptable processes, in: *White Paper der IBM Corporation, Rational Software 2007,* Nr. S. 1 - 4
22. Karlsson, F. and Agerfalk, P. J. (2004) Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets, *Information and Software Technology* (46:9), pp. 619-633.
23. Kuhnen, L. (2009) Demand for e-tools in transport and trade - A customs' perspective, in *United Nations Economic Commission for Europe, Joint Trade and Transport Conference on the Impact of Globalization on Transport, Logistics and Trade: The UNECE Work - United Nations Economic Commission for Europe*, Geneva, Switzerland.
24. Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice - Findings from Early SOA Implementations, in Hubert Österle, Joachim Schelp and Robert Winter (eds.) *Proceedings of the 15th European Conference on Information Systems (ECIS)*, St. Gallen, Switzerland, pp. 1643-1654.
25. Legner, C. and Vogel, T. (2008) Leveraging Web Services for Implementing Vertical Industry Standards: A Model for Service-based Interoperability, *Electronic Markets* (18:1), pp. 39-52.
26. Legner, C. and Wende, K. (2007) The Challenges of Inter-organizational Business Process Design – a Research Agenda, in Hubert Österle, Joachim Schelp and Robert Winter (eds.) *Proceedings of the 15th European Conference on Information Systems (ECIS)*, pp. 106-118.
27. Leist, S. and Zellner, G. (2006) Evaluation of Current Architecture Frameworks, in Lorie M. Liebrock (ed.) *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC2006)*, ACM Press, 03.04.2006, New York, pp. 1546-1553.
28. March, S. T. and Smith, G. F. (1995) Design and Natural Science Research on Information Technology, *Decision Support Systems* (15:4), pp. 251-266.
29. Motahari Nezhad, H. R., Benatallah, B., Martens, A., Curbera, F. and Casati, F. (2007) Semi-Automated Adaptation of Service Interfaces, *Proceedings of the 16th International World Wide Web Conference 2007*, pp. 993-1002.
30. OASIS (2002) Business Transaction Protocol, Organization for the Advancement of Structured Information Systems.
31. Papazoglou, M. (2008) Web Services: Principles and Technology, Addison-Wesley Verlag, Harlow England.
32. Pautasso, C., Zimmermann, O. and Leymann, F. (2008) RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision, *Proceedings of the 17th International Conference on World Wide Web, Beijing, China*, ACM, pp. 805-814.

33. Ratnasingam, P. (2004) The impact of collaborative commerce and trust in web services, *Journal of Enterprise Information Management* (17:5), pp. 382-387.
34. Rossi, M. and Sein, M. K. (2003) Design research workshop: A proactive research approach, *Paper presented at the Twenty-Sixth Information Systems Research Seminar in Scandinavia*, Haikko, Finland.
35. Singh, M. P. and Huhns, M. N. (2005) Service-oriented computing: semantics, processes, agents, John Wiley and Sons, West Sussex, England.
36. Umapathy, K. and Purao, S. (2007) A theoretical investigation of the emerging standards for web services, *Information Systems Frontier* (9:1), pp. 119-134.
37. UN/CEFACT (2006) XML Naming and Design Rules, Version 2.0, United Nations Centre for Trade Facilitation and Electronic Business, Geneva, Switzerland.
38. UN/CEFACT (2009) UN/CEFACT's Modeling Methodology (UMM): UMM Meta Model – Foundation Module Candidate for 2.0, United Nations Centre for Trade Facilitation and Electronic Business, Geneva, Switzerland.
39. Vidgen, R., Francis, D., Powell, P. and Woerndl, M. (2004) Web service business transformation: collaborative commerce opportunities in SMEs, *Journal of Enterprise Information Management* (17:5), pp. 372-381.