The Role of Services in Governmental Enterprise Architectures – The Case of the German Federal Government

Dominik Birkmeier[*], Sabine Buckl[†], Andreas Gehlert[‡], Florian Matthes[†], Christian Neubert[†], Sven Overhage[*], Sascha Roth[†], Christian M. Schweda[†], Klaus Turowski[*]

[*]University of Augsburg
Business Informatics and Systems Engineering
Universitätsstr. 16, 86159 Augsburg, Germany
{dominik.birkmeier, sven.overhage, klaus.turowski}@wiwi.uni-augsburg.de

[†]Technische Universität München
Software Engineering for Business Information Systems (sebis),
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl, matthes, christian.neubert, sascha.roth, christian.m.schweda}@mytum.de
http://wwwmatthes.in.tum.de

[‡]Federal Ministry of the Interior, Germany
andreas.gehlert@bmi.bund.de

ABSTRACT

In the public sector, information technology (IT) as means to support governmental processes is as important as in industry today. Delivering high quality eGovernment services requires an efficient and effective IT support. This IT support can only be provided if the requirements specified in the processes are correctly and completely transformed into IT solutions. Services are seen as major means to support this transformation. In this chapter, we propose a method which systematically translates business processes into services. The method contains 1) a data model describing the structure of the work products of the method, 2) a technique for emergent data modeling, which allows its users to customize the data model according to the government's needs, 3) a role model describing the required competencies for each step, and 4) a process model describing the required steps to derive services from business processes. To succeed in a governmental context with diverse, federative organizational structures, the method needs a high degree of flexibility. In particular, the proposed method has been designed to be compatible with different process modeling techniques.

1 INTRODUCTION

Electronic Government (eGovernment) has a long tradition in Europe. This long tradition was recently underlined by the Ministerial Declaration on eGovernment (Ministers of the European Union, 2009). Among others, this so-called Malmö declaration strives for designing eGovernment services around the needs of the users, to reduce the effort for using these services and to increase the availability of public sector information (p. 2f).

The Malmö declaration was also influenced by Europe's Digital Agenda (European Commission, 2010c). The Digital Agenda describes problem areas, political goals and actions for the development of Europe's IT. The major elements of Europe's Digital Agenda are the notions of business process orientation and service orientation (p. 15). Business process orientation as well as service orientation have been refined in more technical terms in the European Interoperability Strategy (EIS, European Commission, 2010a) and the European Interoperability Architecture (EIF 2.0, European Commission, 2010b).

Especially the EIF emphasizes the fact that eGovernment is more than the communication between administrations and citizens (A2C) or the communication between administrations and businesses (A2B). It particularly includes the communication between different administrative bodies (A2A). Although this communication is “invisible” to the citizen and the business, it directly supports the goals of increasing the efficiency and effectiveness of public services as expressed in the Malmö Declaration and the EIS (European Commission, 2010a; Ministers of the European Union, 2009). Therefore, we understand eGovernment as follows:

> *eGovernment is the IT-supported exchange of services between public administrations and citizens (A2C), between public administrations and industry (A2B), and between different public administrations (A2A).*

Providing such administrative services efficiently requires that these services are supported by IT. The IT support, however, is only effective if the requirements of the business processes are correctly and completely translated into IT solutions. In this chapter, we cover the first step of this translation: We propose a method to systematically derive services from business processes.

Research Question: How can services systematically be derived from business processes?

Thereby, we understand the term *service* as follows:

> *A service is a set of requirements, which is already supported by IT solutions or will be realized by IT solutions in the future. By IT solutions we mean any software, or component thereof, which is capable to realize a service.*

Given this definition, services are the crucial link between business and IT. On the one hand services are extracted from business processes and are directly linked to them. On the other hand IT solutions may implement one or more services so that these IT solutions are also linked to services. Services enable the business process engineer to support his/her processes with IT without any knowledge of the internal structure of the IT solutions. In the other direction, the solution owner does not need to have complete knowledge of the business processes to provide IT solutions, which are useful for the business.

Although a sound method is required to derive services from business processes systematically, such a method needs to respect the organizational settings of government agencies. The German Constitution for instance prescribes that every federal ministry is independent (§65 "Grundgesetz für die Bundesrepublik Deutschland," 1949). Consequently, it is very difficult to establish a certain technique in the entire German Federal Government. Therefore, a sound method needs to be flexible enough to respect a divergent degree of formalism as input. In our case, it means that the proposed method should be compatible with many process modeling techniques and it should be possible to tailor the method according to the needs of the government agency.

We follow a design science research approach to provide the depicted method. The method is described in seven sections: In Section 2 we elicit requirements for the envisioned method. Based on these requirements, we discuss related work in the area of service identification and knowledge management via Web 2.0 and Enterprise 2.0 tools in Section 3. The method itself is introduced in Section 4. Thereby, we demonstrate how the “wisdom of the crowds” (Surowiecki, 2004) can be used to enable enterprise architecture (EA) management. By using Web 2.0 techniques, wikis, and an open templating mechanism, we show how the ivory tower syndrome can be cured, typical pitfalls are avoided, and employees are empowered to contribute their expert knowledge. Section 5 applies the method to an example from the governmental domain. It serves as an evaluation of the proposed method. Section 6 describes future directions that we see for the proposed method and Section 7 summarizes our findings.

2 REQUIREMENTS

Industry trends such as globalization, rapid economic change, and the necessity to foster an organization's sustainable competitive advantage (Wagter, van den Berg, Luijpers, & van Steenbergen, 2005) have also reached public administrations today. One of the main drivers for innovation in public administrations is the need to provide governmental services in a digital form across different government agencies and different countries at low costs and without barriers (IT Planning Council, 2010). This requires on the one hand interoperable systems and on the other hand systems that can be modified quickly to comply with future jurisdiction.

A method for systematically deriving services is a key prerequisite for realizing eGovernment. This method should contain all constituents of a method as defined in the method engineering discipline (Brinkkemper, Saeki, & Harmsen, 1999):

> Req1 The method for deriving services from business processes must contain a *process model* explaining important activities on how services should be derived; a *role model* explaining the roles, their competencies and responsibilities for certain activities in the process model; and a *data model* explaining how services are described.

Since the method should be applicable in many governmental agencies, the envisioned method especially cannot be based on a specific process modeling technique such as the Business Process Modeling Notation (BPMN). Therefore, we assume:

> A1 The method for deriving services from business processes cannot rely on a single business process modeling technique.

Assumption A1, however, does not exclude the possibility to use conceptualizations common to many process modeling techniques such as activity or control flows in the envisioned method.

Governments usually have a strong separation of business and IT units. People working in these units have also very different backgrounds and qualifications. It is unlikely to find people with a strong IT background in a business unit; or people with a strong business background in the IT units (Frederiksa & van der Weideb, 2006). Since services are at the junction between business and IT, the proposed method must be easily understood by business and IT experts. If the method is too complicated and cannot be communicated to both business and IT-experts the method might not come into use:

> Req2 The envisioned method must be designed in a way so that it can be easily understood by business and IT experts, i.e. it must build on a common terminology.

To reduce costs and to enhance the quality of IT assets, the method needs to support reuse. Therefore, the envisioned method has to provide support for cataloguing IT services and for managing this catalogue:

> Req3 The envisioned method allows adding services to a service catalogue and supports reshaping those services according to already existing services in the catalogue.

To link the different data sources within the organization, a common description for services has to be developed. This development often suffers from the 'ivory-tower syndrome', i.e. leads to the creation of a wish list in which each stakeholder asks for the bits and pieces of data s/he is interested in. This results in an unmaintainable large model describing a service.

Req4 The envisioned method has to develop and/or provide a common service description model understood by stakeholders with different backgrounds.

Information on processes and services has to be maintained on a regular basis to be useful for governing the IT. The documentation process accordingly must be conducted in a way, which is on the one hand feasible for stakeholders with various backgrounds (e.g. process or application owners), and on the other hand shows the benefits of their time spent on sharing knowledge. A concept for information maintenance requires motivating mechanisms for the information providers. Such mechanisms may be managerial orders or financial rewards. Such reward mechanisms are usually not available in the decentralized environment of interoperating administration bodies. In this light, “soft” incentive mechanisms are required, making the utility of the shared information visible to the corresponding provider.

Req5 Information providers must receive feedback on the utility and the appropriateness of the shared information.

In our context, service design targets the reuse of existing services on the basis of functional requirements. Thereby, the functional requirements may be provided in one department and be required in another at a different location. Hence, it is crucial that different departments immediately are enabled 1) to share information about a service and 2) to discover already maintained services. Ideally, so-called base-services are identified which can be provided in a central manner, since those services provide functionality frequently used by multiple departments.

Req6 The envisioned method has to involve stakeholders of different departments distributed over several locations to assure service reusability.

Although we have motivated our requirements from the governmental domain, we believe that these requirements are equally applicable in other industries. The proposed method bridges the gap between business and IT and has only limited requirements on the needed inputs. Therefore, we believe that our chapter is not specific to the governmental domain and contributes also to the knowledge body of EA management where business-IT-alignment and service-orientation is a central topic of interest (Aier, Gleichauf, & Winter, 2011; Buckl, Marliani, Matthes, & Schweda, 2011).

3 BACKGROUND

There are many approaches that address the topics of deriving services or of using the wisdom of the crowds to gather information. However, an approach to combine these two perspectives on information gathering in the context of service modeling is missing. Subsequently, we prepare our solution by investigating the existing knowledge base with respect to different methods and techniques that can be used to derive services (Section 3.1) and by revisiting typical functionality provided by Enterprise 2.0 platforms that has been proven to be useful to gather information from stakeholders with different backgrounds (Section 3.2).

3.1 Approaches for deriving services

Besides service-oriented design in general, especially the derivation of services has frequently been addressed in literature. The existing approaches, however, differ in their support for a systematic procedure. They range from general recommendations that should be considered during the derivation process (SAP, 2005) to approaches which cover at least some or ideally all parts of a comprehensive method as postulated in *Req1* (Aier & Winter, 2009; Azevedo, et al., 2009; Erl, 2005; Erradi, Anand,

& Kulkarni, 2006; Klose, Knackstedt, & Beverungen, 2007; Winkler, 2007), whereas the latter is not achieved yet. The service derivation approaches found in literature consider different aspects of information for the derivation process (such as activities, data, control flows, data flows, etc.). While similar approaches for older paradigms such as component-based development are sometimes dependent on specific modeling techniques (Jain, Chalimeda, Ivaturi, & Reddy, 2001), we found none of the examined service-oriented approaches to be that restrictive (*A1*).

Some approaches build upon a purely technical view of services and consequently apply an analysis of source-codes and database schemes to derive services (Erradi, et al., 2006). Others aim at involving both business and IT experts into the process and, hence, provide a common terminology which can be equally understood by both sides (Azevedo, et al., 2009; Winkler, 2007). However, most approaches fail to address this requirement (*Req2*). With respect to the necessary input, the proposed approaches vary in their consideration of existing services (*Req3*). Only some are able to include existing structures during the derivation process (Erradi, et al., 2006; Klose, et al., 2007; SAP, 2005), while the others lack such possibilities so far.

With respect to the model used for service description (*Req4*), the proposed approaches range from domain-oriented to technically-oriented solutions. None of the approaches aims at defining a common model that is understood by stakeholders with different backgrounds. The understanding of services is a key influence factor on the results of an approach, however (Birkmeier, Klöckner, & Overhage, 2009). To the best of our knowledge, none of the service derivation approaches proposed envisions an incentive mechanism for sharing information (*Req5*) or an explicit integration of stakeholders from different locations, departments, etc. to foster the reuse of services (*Req6*).

	SAP (2005)	**Erl (2005)**	**Erradi et al. (2006)**	**Winkler (2007)**	**Klose et al. (2007)**	**Azevedo et al. (2009)**	**Aier & Winter (2009)**
Provisioning of process, role, and data models (*Req1*)	(×,×,×)	(✓,×,✓)	(✓,×,✓)	(✓,×,×)	(✓,×,×)	(✓,×,×)	(✓,×,✓)
Independency from modeling techniques (*A1*)	✓	✓	✓	✓	✓	✓	✓
Common terminology for business and IT (*Req2*)	×	×	×	✓	×	✓	×
Consideration of existing services (*Req3*)	✓	×	✓	×	✓	×	×
Common model (*Req4*)	Technically-oriented	Technically-oriented	Domain-oriented	Domain-oriented	Domain-oriented	Domain-oriented	Domain-oriented
Information provider feedback (*Req5*)	×	×	×	×	×	×	×
Stakeholder involvement (*Req6*)	×	×	×	×	×	×	×

Table 1: Summary of related approaches in service derivation

A detailed comparison of the different approaches is summarized in Table 1. As it turns out, none of the approaches is able to fulfill all requirements. This observation is also supported by extensive literature studies on the state of the art in service derivation, which draw similar conclusions and attest that additional research effort is required to create more mature approaches (Birkmeier, et al., 2009; Dietz, Juhrisch, & Grossmann, 2011; Kohlborn, Korthaus, Chan, & Rosemann, 2009).

3.2 Web 2.0 & Enterprise 2.0

The term Web 2.0 has increasingly gained attention in the last years. In Tim O'Reilly's definition, Web 2.0 terms modern applications facilitating interactive collaboration and communication via the Internet (O'Reilly, 2008). He puts emphasis on Web 2.0 as applications which enable users to immediately share and reuse information. Since Web 2.0 applications primarily focus on users' personal reputation and expertise the term is often used as a synonym for active user participation in the Internet. Objects in Web 2.0 applications which are primarily created by the users themselves and not statically given by a web provider (e.g. media objects such as videos) are termed as "user-generated content" (O'Reilly, 2008).

Surowiecki (2004) introduces the term "the wisdom of crowds". This principle means that the quality of decisions conjointly taken by a group is often better than the one of those taken by particular persons. This phenomenon especially applies for Web 2.0 applications since they mainly support collaboration and communication tasks in teams and groups.

Today's most prominent Web 2.0 application in which the principles "the wisdom of crowd" and "user-generated content" (O'Reilly, 2008) are successfully applied, is the Wikipedia Encyclopedia (Leuf, 2001). This project aims to collect the world's knowledge, whereby everyone can contribute. A wiki is a "website" that allows the creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG text editor"[1]. In contrast to a classical content management system, where changes of the content must go through an editorial process before they are shown on the website, changes in wikis are immediately visible. An information consumer can thus instantly switch to information provisioning, making him or her effectively what is called a "prosumer" (Chang, 2006).

McAfee (2005) describes the application of Web 2.0 techniques in enterprises as so-called Enterprise 2.0 techniques. A great number of software vendors combine different individual Web 2.0 solutions to integrated Enterprise 2.0 platforms. Besides delivering the advantages of classical Web 2.0 applications, e.g. ease of use, these platforms are especially optimized for the deployment in enterprises, e.g. by means of advanced access control lists and desktop-oriented user interfaces.

In Büchner et al. (2009), the platforms of leading Enterprise 2.0 vendors are compared to each other. The authors provide a detailed functional analysis of the platforms and describe the

- supported *content objects* which contain user generated content, e.g. wikis and blogs, and the
- provided *Enterprise 2.0 services* which are operations on the content objects, e.g. tagging.

In the following, we discuss selected Enterprise 2.0 services applied to wiki-based content and detail their usability in our application context:

S1: Authoring services support the users during the collaborative creation and manipulation of wiki-pages. These pages combine unstructured information, e.g. plain text, links and images, with semi-structured content, e.g. attribute-value pairs. Semi-structured content can further be organized into *templates* that define the attributes for a more specific type of content, e.g. a wiki-page describing a business process.

S2: Tagging services support the collaborative categorization of content objects. A tag is a keyword that categorizes a content object against one or more user-created classification schemas (Golder & Huberman, 2005). More sophisticated implementations of tagging services facilitate to link certain tags, called "type tags", to templates for semi-structured content.

S3: Search services can be used to find content objects fulfilling specified criteria. These criteria can target the full text of the wiki-pages and can access semi-structured content as well. In particular, the user can specify searches that find pages, which supply a specific value for a chosen attribute. Furthermore, the searches can be restricted to deliver only wiki-pages tagged with selected keywords.

[1] http://en.wikipedia.org, visited on May 10th, 2011.

S4: Link management services support the users in creating and maintaining references between wiki-pages. Internal links, i.e. links to content objects managed by the platform itself, are updated by the link management, whenever the link target is re-named. Thereby, consistency of the references is ensured. Link management services further highlight links that reference no longer existing content objects. Further, link management can be used to restrict the valid values for an attribute-value pair to wiki-pages that supply a specific type-tag.

S5: Awareness & Feedback services help the users to follow the activities of other users. Users can define watch-lists in order to get informed, when the content of selected wiki-pages is changed. Change feeds provide an overview about the ongoing editing activities in the wiki. View trackers anonymously log the visits of selected wiki-pages. Feedback mechanisms, such as comments and ratings, supply means to express the opinion on selected content objects.

In Section 4.2 we apply the aforementioned Enterprise 2.0 services in a technique for emergent data modeling.

3.3 Conclusions from the Literature Analysis

From the literature we reviewed we can conclude that no method exists that fulfills our requirements completely. The strengths of the service-oriented approaches are their level of completeness and their formality. For instance, all reviewed approaches are independent from the modeling language used for process modeling and most approaches have at least two of the required parts of a modeling method. However, the service-oriented approaches lack the collaboration aspect. None of the analyzed approaches supports feedback to the information provider or fosters the involvement of stakeholders with different backgrounds in the service identification task. The Web 2.0 and Enterprise 2.0 approaches focus mainly on collaboration but do not provide us with (semi-) formal methods on how to derive services from business processes.

The results of the literature analysis mean that the enterprise architect has either well-defined methods available that come with the risk of not being used in the enterprise since they lack appropriate collaboration techniques; or s/he uses well-established collaboration techniques without the necessary tools and guidance on how to derive services from business processes. Therefore, the method developed in Section 4 combines the two worlds:

- the proposed method is formal enough to allow the enterprise architect to derive services from business processes;
- the proposed method has very low requirements for its inputs and is, therefore, easy to use;
- the proposed method incorporates Web 2.0 techniques to foster collaboration; and
- the proposed method ensures feedback to the information provider to motivate him/her to provide accurate and up-to-date information in the future.

4 A METHOD FOR DERIVING SERVICES FROM BUSINESS PROCESSES

This section describes a method for deriving services from business processes along the following method constituents:

- *Data Model*: The data model describes the structure of the documentation (Section 4.1).

- *Technique for Emergent Data Modeling*: The technique describes how to adapt and extend the data model based on information gathered from different stakeholders (Section 4.2).

- *Role Model*: The role model describes the roles with their competencies and responsibilities for the activities in the process (Section 4.3)

- *Process Model*: The process model describes the relevant activities of the method (Section 4.4)

Each of the following subsections may contain specific assumptions, which restricts the design space (Gehlert, Schermann, Pohl, & Krcmar, 2009; Schermann, Gehlert, Krcmar, & Pohl, 2009). These assumptions and their implications are discussed in Section 6.

4.1 Data Model

The data model describes the general structure of the work-products of the method. It is depicted in Figure 1. Key elements of the data model originate from the business and the IT side (cf. *Req2*) and are: *business process* and *activity* (on the business side) and *services* (on the IT side). The *function* is the linking element between the two worlds.

[Image „Figure1-Metamodell-eGovPaper-2-1.tif“ here]

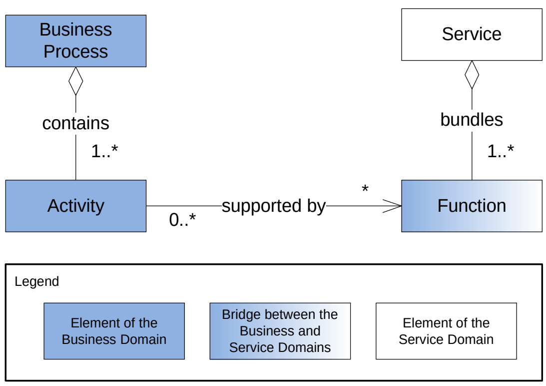

Figure 1: Data model of the proposed method (modeled as UML class diagram)

Assumption A1 prescribes that the method cannot rely on a concrete modeling technique. However, for the data model we need to assume that business processes do exist and that these business processes consist of activities. Thus, we can formulate the following assumption:

A2 The results of the business process analysis are business processes, which consist of activities.

Assumption A2 means that the proposed method can only be applied *after* a business process analysis (or during its later phases). It also decouples the application of the method from business process modeling and particularly from a concrete process modeling technique. Activities are central concepts of many process modeling techniques, e.g. the Event-Driven Process Chain (EPC), the Activity Diagrams (AD), and the Business Process Modeling Notation (BPMN, Patig & Casanova-Brito, 2011). Therefore, assumption A2 is suitable for a governmental setting without a prevalent process modeling technique.

Activities are the basis for functions. A *function* represents a functional requirement described in the language of the process modeling expert. It is the bridge between the business and the IT worlds. Activities and functions share a many-to-many relationship. This many-to-many relationship decouples the structures of the business processes from the services and fosters a loose coupling between those two elements. The relationship between activity and function can be described as follows:

- An activity may be *manual* and may not need IT support. Therefore, the activity may not have any function assigned.
- An activity may have exactly one function assigned. The activity is *automated*.
- An activity may have more than one function assigned. There are two reasons for this situation: either the business process is more coarse-grained than the functions or there is the need for an additional IT support for this activity. We call the functions of the latter case *implicit functions*. They result from the fact that not all IT-related aspects are reflected in the business process. For example, login or security functions are usually not modeled in business processes.
- A function may be assigned to more than on activity. This indicates that the business process is more fine-grained than the functions. This is typically the case when data objects are created, modified or quality-assured. These activities may well be represented separately in a business process. In IT, this situation may just be represented as one function, which modifies a set of attributes of a particular data object.

Services bundle similar functional requirements and thus functions, which are or should be implemented in IT. As a service should fulfill a purpose, it contains at least one function. The collection of all services is called *service catalogue*.

4.2 A Technique to Emergent Data Modeling

For refining the model discussed above, we introduce a technique that builds upon Enterprise 2.0 services (cf. Section 3.2) to create data models over time using structured content in an Enterprise 2.0 platform incorporating basic thoughts of Wiki-based systems. While the common process for developing models is based on “schema first”, data second, our approach focuses on “data first, schema second”. The technique is facilitated by the principles “the wisdom of crowds”, “active participation” and “user-generated content” as introduced in Section 3.2.

The Enterprise 2.0 service S1 provides a mechanism known as *auto-completion*, which recommends values and names already used for the corresponding attribute in the semi-structured content. This mechanism is a key component for the technique to emergent data modeling, as the recommendations facilitate the development of a consistent terminology. In particular the accidental introduction of new concepts and terms by occasional typographic errors is avoided.

Moreover S1 additionally provides mechanisms to explicitly specify the type of a wiki page, e.g. by means of wiki templates or tags. The type of a page indicates the class of the object being described in the content of the wiki page. A great number of Wiki-based systems are using types (or concepts which are similar to types, e.g. templates) to enable authors to reuse often needed structures as well as to define specific integrity constraints. For example a wiki template for a town could specify that all town instances (i.e., pages which are using the template “town”) should provide an attribute ‘Population’. In the context of data modeling, the wiki templates can be regarded as instantiations of the ‘construct templates’ introduced in the ISO Standard 19440 (ISO/IEC, 2007). Such templates supply the name of a modeling type, describe the properties as well as relationships of this type, and textually define the semantics of the type. Multiple Wiki-based systems offer functions to provide textual descriptions along the templates, which can be used to supply a semantics definition. In Matthes et al. (2011), so-called type tags are used for typing a wiki page instead of using templates. Furthermore, this approach introduces mechanisms providing a smooth transition from unstructured textual content to more structured wiki pages.

Besides the auto-completion mechanism for attribute names and attribute values, S1 provides a further recommendation technique, namely *attribute suggestions*. Attribute suggestions are generated based on a statistical analysis (cf. S3) of frequently used combinations of tags (cf. S2) and attributes in Wiki-based systems. The name of the recommended attribute is shown with an empty value field (i.e. an attribute stub) on similar wiki pages to urge the wiki authors to provide a value for this empty field. For instance, if a particular wiki page is tagged with the keyword ‘business process’ and additionally provides an attribute “acronym” on other wiki pages also tagged as ‘business process’ a stub (attribute suggestion) for the attribute “acronym” is shown. One the one hand, attribute suggestions facilitate a data-driven evolution of the data model, on the other hand they contribute to a consistent terminology and a uniform data model, similar to the auto-completion mechanism.

In some Wiki-based systems, types have an impact on both the auto-completion as well as on the attribute suggestions. For instance, if it is specified (in the type or template) that the aforementioned attribute ‘Population’ may only consist of integer values, the auto-completion control can provide input support optimized for integers. Furthermore, constraints can also have an impact on the ranking of auto-completion result lists. For instance, if an attribute (e.g. constituent country) is constrained to link values referencing wiki pages with a specific type (e.g. country) the auto-completion mechanism prefers pages fulfilling this constraint.

Changes to the type may also influence the attribute suggestion mechanism. For example, in case of the definition of an additional attribute[2] on the type level, an attribute suggestion is provided on pages according to this type. By doing so, decisions made by a schema designer (data model designer) are immediately visible for the wiki page authors. Thus, the designers and authors enter into dialog and the evolution of the data model is facilitated and the set of terms converges to a commonly accepted terminology.

Kurpjuweit and Winter (2007) explain that the relations between objects are more important than the particular properties of the object itself. Therefore, Wiki-based systems commonly provide services (cf. S4) to create (hyper-) links to other wikis pages as quickly and efficiently as possible. For instance, in some wikis a service is provided to transfer plain attribute text values to hyperlinks with little effort (i.e. with on-click). Since in this case plain text is transferred to an object having an individual identity (URL), we call this mechanism *objectification*. For users, optimized objectification mechanisms are very powerful to facilitate the evolution of the data model.

The introduced technique (mechanisms and services) enable data modeling from two perspectives:

- Bottom-up, i.e. the data model emerges spontaneously due to the interplay of particular wiki pages and their structured elements.
- Top-down, i.e. the definition of the data model takes place on a meta-level independent of the particular wiki pages.

In Chapter 5, we explain how this technique brings benefits to our project context and contributes to the fulfillment of the requirements as specified in Section 2. Furthermore we describe how both methods (bottom-up and top-down data modeling) interlock and thereby benefit from each other.

4.3 Role Model

The method operates at the junction of the business and IT worlds. The role model reflects this fact and introduces roles with dedicated responsibilities in each world:

- *Process Engineer*: The Process Engineer is responsible for designing business processes. S/he knows the processes of the organizational unit in focus and is able to express those processes

[2] “Additional” means that no particular wiki page already provides ~~an~~ a corresponding attribute.

in a modeling language, which distinguishes at least processes and activities (cf. A2). Additionally, the role includes the rights to (re-) design the business processes of the organizational unit and to decide, which activities should be supported by IT. Furthermore, the Process Engineer is knowledgeable about organizational aspects, which are not covered in the business process and which may lead to implicit functions.

- *Service Engineer*: The Service Engineer is responsible for the service catalogue. S/he knows the structure and the content of the services in the service catalogue. In particular, the Service Engineer should know which services are already realized by IT. Furthermore, the Service Engineer is empowered to decide, which services may be added to the service catalogue and has experience in identifying implicit functions and in identifying services. S/he has in-depth knowledge of the proposed method for deriving services from business processes.
- *Stakeholders*: The Stakeholders may be different groups of professionals actually executing an activity of a process. They have insights regarding these activities and the corresponding functions from day-to-day application thereof in their professional occupations.
- *Solution Owners*: The Solution Owners host implemented service realizations, e.g. a running business application offering electronic publishing services.

In addition to the roles described before, it would be desirable that the Process Engineer has some knowledge of the principle of service-orientation and the method for deriving services. Furthermore, the Service Engineer should have some knowledge of business process modeling and the domain of the project. This additional knowledge will ease the communication between the two worlds.

4.4 Process Model

The presented method derives services from business processes, which may be – according to assumption A1 – described in an arbitrary format. This format can range from formal process modeling techniques to informal textual descriptions. As a consequence, the business process descriptions may not state the information and data on which the processes operate. Therefore, many existing service identification approaches cannot be applied in such a setting.

A key complexity in the process-based identification approach is the consequence of the level of abstraction on which the activities constituting the business processes are documented. This level of abstraction is determined by the Stakeholders' understanding of the business and does not necessarily match the level of abstraction, on which discussions on functions take place. In particular, two types of mismatch have to be distinguished:

- Process abstractions, where relevant IT functions, such as authentication or encryption, are only implicitly alluded to.
- Functional abstractions, where different activities can be supported by the same underlying function, e.g. apply this function only on different information.

The method for deriving services from business processes has to account for these types of mismatches. Figure 2 gives an overview on the method's underlying process.

[Image „Figure2-overview.tif“ here]

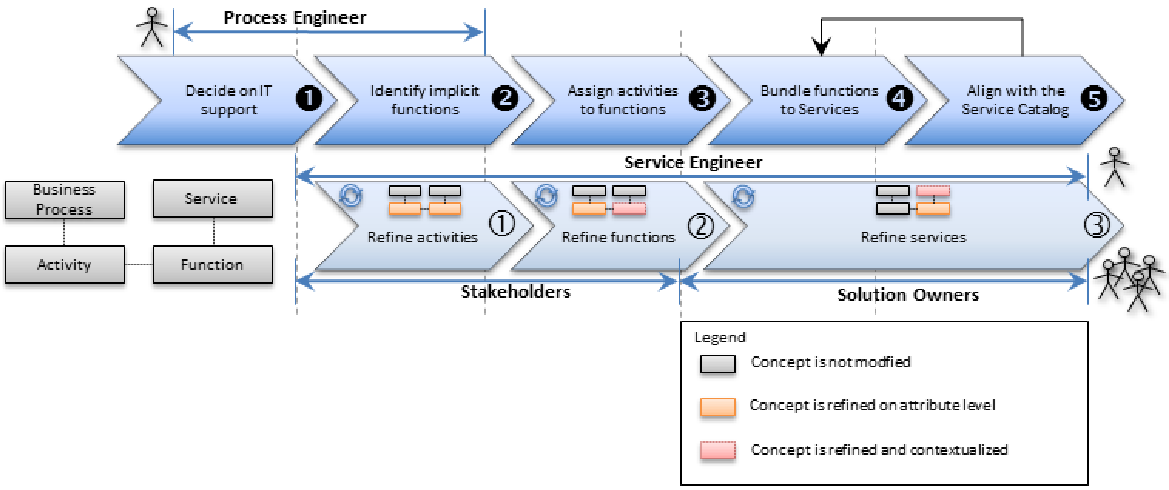

Figure 2: A process for deriving services from business processes

In the first step of the method (step ❶ in Figure 2), the Process Engineer identifies the activities and decides which activities should be supported by IT. The Service Engineer may support the Process Engineer with information about already existing services, which may completely or partially satisfy the business needs. In addition, the Service Engineer may explain to the Process Engineer that the realization of certain function is not yet feasible. In this situation, Process Engineer and Service Engineer may discuss a compromise solution.

In the next step (step ❷ in Figure 2) the Process and Service Engineers jointly identify implicit functions, which are not described in the activities of the business process. This is the case when the business process is more abstract than the functions. This situation also covers functions needed for an IT support of the process (e.g. authentication, authorization, encryption, etc.) which are usually not modeled during the business process analysis. The Service Engineer contributes these functions based on the experiences of frequently used implicit functions.

During the identification of the implicit functions, the Process and the Service Engineers rely on the knowledge of the professionals, i.e. stakeholders, actually using the explicit function. These stakeholders are involved and collaboratively refine the description of both explicit and implicit functions using the technique for emergent data modeling (*Req4*). In particular, new attributes are defined (*Req6*) and corresponding values are supplied, further specifying the nature of the discussed activities (supporting activity ① in Figure 2).

In the subsequent step (step ❸ in Figure 2), activities which require the same function are linked to the same functional requirement. This is the case when the business process is less abstract than the functions. An example of this situation is the creation, modification, authorization and submission of a document. These activities may be relevant from a business process perspective. However, in IT these processes may modify only a set of attributes of a document in a document management system. Therefore, there is only one functional requirement “change attributes of a document” needed to support this activity. Step ❸ is also relevant for implicit functions. It may, for instance, be necessary that the user of the later system is authorized to execute a particular step in the business process. Again, the “authorization” function is assigned to more than one activity in this case.

After the execution of step ❸ of the proposed method, the service and process engineers have identified a set of functions and have assigned those functions to the activities of the business process. Therefore, we need to assume:

A3 All implicit functions are known or can be identified when deriving services from processes.

Assumption A3 says that implicit functions are an additional input to the proposed method.

The process and service engineers involved in steps ❷ and ❸ need to understand the functions and activities in the business process correctly. If different interpretations of functions and/or activities persist, the resulting functions may not be correctly assigned to activities and, therefore, the support for the business process may not be optimal. Therefore, steps ❷ and ❸ should be supported by a collaboration tool, which facilitates the development of a concise and consistent understanding of the documented functions and activities. In such a tool, the Service Engineer can propose an assignment of the different activities to corresponding functions and therein refine the description of the function appropriately. The stakeholders performing the corresponding activities receive notifications on the assignment of "their" activities (*Req6*). They can subsequently provide additional information on the nature of the activity. Further, they may relate standardization documents, relevant guidelines or white-papers to the function and detail the function specification as required (supporting activity ② in Figure 2). Feedback mechanisms facilitate stakeholder discussions on the details of the function (*Req5*). In case of an erroneous assignment, the stakeholders may enter direct interaction with the Service Engineer.

Steps ❹ and ❺, see Figure 2, are executed iteratively. They both target the bundling, i.e. the aggregation, of technically related functions to services. The bundling of related functions to services creates service demands which have to be aligned with an existing service catalogue. This catalogue contains initial descriptions of the available services. Comparing the service demands resulting from the bundling of functions with the available services, the Service Engineer may identify the following types of alignment:

1) *Full service alignment*: The identified service demand fits exactly with an existing service of the service catalogue. Existing IT solutions assigned to that service can directly be reused.
2) *Full function alignment*: In this case, all functions of the identified service demand are represented in the service catalogue but the functions may belong to different services. To foster reuse, the service demand should be re-organized according to the services of the service catalogue. As in situation 1), existing IT solutions can be directly reused.
3) *Partial function alignment:* In this case, only parts of the functions of the identified service demand are represented by services in the catalogue. The recommendation in this situation would be to re-organize the service demand: the set of functions covered by the service catalogue should be bundled to one service; while the rest of the functions should be bundled to another service. In this way, existing IT solutions can be partially reused while the missing functions need to be implemented separately.
4) *Complete function mismatch:* In this case, none of the functions are covered by services in the catalogue. A decision must be made, whether the service should be added to the service catalogue or not. This situation will most likely appear during the initial set-up of the service catalogue.

The relevant analysis is supported by Enterprise 2.0 techniques, especially the *Search* service allowing to identify possibly matching services based on attributes but also based on full-text descriptions.

Situations 2) and 3) require splitting an already defined service. This may indicate that the service catalogue and the identified services are structured differently. Therefore, the identification of services (step ❹ in Figure 2) should be repeated iteratively. Additionally, in situation 3), the functions, which are not supported by a service of the catalogue, need to be bundled and new services need to be identified. This will most likely lead to situation 4) in the next iteration of the service identification process. During the analysis of the alignment of the service demands with the services in the catalogue, the Service Engineer may refine the description of the demanded service as well as that of an existing service (supporting activity ③ in Figure 2). In particular, new characterizations for non-functional requirements, e.g. security requirements, may be added to the service description template. The Solution Owners of the services' underlying IT solutions are in turn notified about the changes and refinements in "their" service descriptions. In case a novel characterization of the services does not match the characterization of the underlying IT solution, the Solution Owners can raise objections to the Service Engineer. In a subsequent iteration of the step, the Service Engineer can then re-align the functions according to the mismatch in non-functional characterizations.

5 ILLUSTRATIVE EXAMPLE SUPPORTED BY ENTERPRISE 2.0 SERVICES

In the following, the proposed method is applied to a typical example of the public sector: the procurement of goods and services. The method steps (Steps ❶ to ❺ in Figure 2) will be exemplified. Concurrently to those method steps, Stakeholders, Service Engineer, and Solution Owners work on a close basis in an iterative manner addressing *Req6*. In addition to the example, the interplay of this cooperative work illustrated in Figure 2 (sub-method steps ① to ③ in Figure 2) will be described subsequently highlighting how concrete Enterprise 2.0 services of the Tricia platform are employed, to support and facilitate a technique for emergent data modeling (cf. Section 4.2).

The business process can be described as follows:

> *The business process starts with a request for needed goods or services. More specific requirements of these goods or services are formulated and a call for bids is issued. The bids are collected and evaluated according to the given requirements. Once the decision is made, one of the bids is accepted and a contract is issued. The goods or services are bought based on this contract. Finally, all relevant documents are archived. Figure 3 shows the resulting process on the left.*

[Image „Figure3-Metamodell-eGovPaper-2-2.tif“ here]

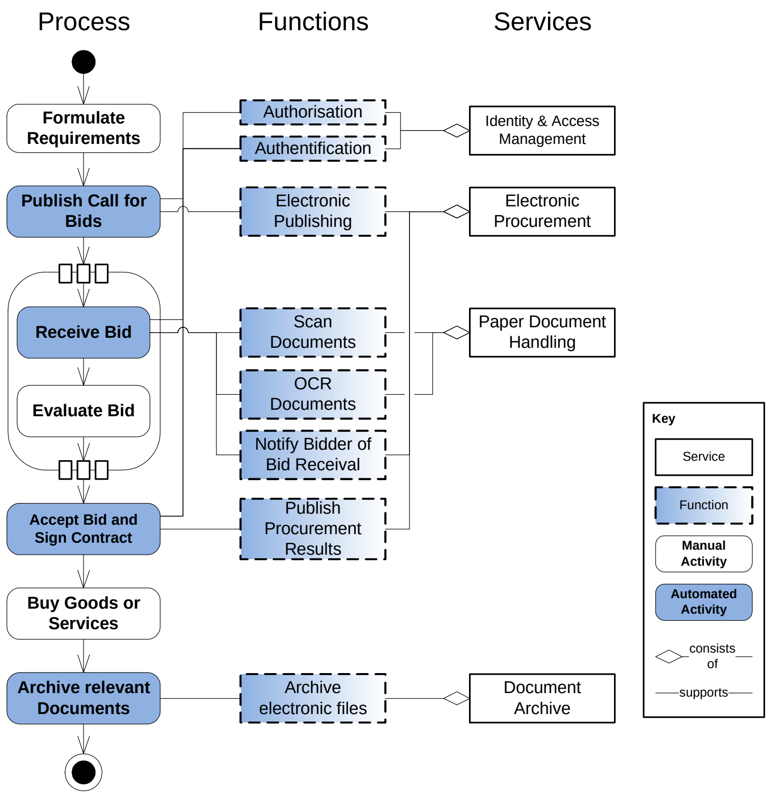

Figure 3: Procurement process with functions and services

The Process Engineer decides on the IT support of the activities in the business process: The activities “publish call for bids”, “receive bid”, “accept bid and sign contract”, and “archive relevant documents” should be automated. IT support is not required for all remaining activities.

According to our process and role models (cf. step ❶ in Figure 2), the Process and the Service Engineers need to agree on the functions behind those activities, which should be automated. The Process Engineer knows that bids are typically received as paper documents. Therefore, s/he proposes to scan those documents so that the documents can be handled electronically. The Service Engineer introduces an optical character recognition (OCR) function to make the documents searchable to ease the evaluation of the bids in the next step. Both the Process and the Service Engineers have to ensure that *implicit functions are identified* and made explicit (cf. step ❷ in Figure 2). The Process Engineer identifies an implicit function: the notification of the bidders once the bid was received. This function should also be supported by IT.

> Based on the activity descriptions of Stakeholders, which may be provided in full text, the Process and Service Engineers are provided with an information base for deciding whether or not certain functions may be supported by IT. These method steps (cf. steps ❷ and ① in Figure 2) can be executed concurrently, due to the employment of an Enterprise 2.0 platform (cf. *Authoring* in Section 3.2). This means: while the Stakeholders and Service Engineer *refine activities*, the Service and Process Engineers jointly identify potential implicitly described functions and make them explicit. In addition, the emergent data modeling technique introduced in Section 4.2 enables the Service Engineer to iteratively extend the model by attributes without explicitly changing the model, e.g. an attribute called *IT-supported* indicating whether or not a certain kind of function can be supported by IT (cf. Section 4.2).

The Process and Service Engineers identify the need for an authorization for the activities “publish call for bids”; “receive bid” and “accept bid and sign contract” (implicit functions; cf. step ❷ in Figure 2). However, both parties agree that the authentication and authorization required in these three activities are identical. Thus, the service engineer decides to assign these three activities to the authentication and authorization functions (step ❸ in Figure 2). While the Service Engineer is concerned with the method step *assign activities to functions* (cf. step ❸ in Figure 2), the Stakeholders are enabled to refine 1) the description of functions and 2) the structure of the functions (cf. step ② in Figure 2). Thereby, Stakeholders can describe functions in full-text, or attribute value pairs employing techniques like e.g. auto-completion with attribute suggestion (cf. Section 4.2 and Figure 4).

In the next step (cf. step ❹ in Figure 2), the Service Engineer bundles the six functions to services. The Service Engineer knows that there are already services in place, which allow handling paper documents. So the functions “Scan Documents” and “OCR documents” are bundled to one service. The same holds true for the function “Archive Electronic Files”. The Service Engineer is not aware of an existing service for the remaining functions. However, the Service Engineer concludes that the remaining functions logically belong together and bundles them to the “Electronic Procurement” service. Figure 4 illustrates how the Enterprise 2.0 platform Tricia supports *bundling of functions to a service* via the Enterprise 2.0 service *Authoring* with auto-completion.

[Image „Figure4-auto-completion.tif“ here]

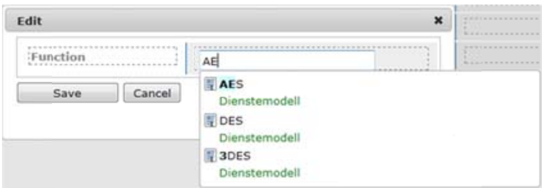

Figure 4: Bundling functions to a service using auto completion

Within the service description (Figure 5), Stakeholders and Service Engineers may specify Tags (cf. *S2* in Section 3.2) for a particular service (cf. ① in Figure 5), so that services 1) can be specified and 2) may be easily found. A full-text description (cf. ② in Figure 5) can be used to describe the service. Solution Owners can also contribute their knowledge on services already realized by IT (cf. ③ in Figure 2). Thereby the Authoring concept ensures that common understanding evolves over time. Moreover, structured information can also be captured, such that explicitly known functions can be referenced and refined, if needed. Therefore, so-called *Type-Tags* (cf. ③ in Figure 5) can be used to specify, which concept is described. In this particular case, referencing functions is technically speaking a multi-value attribute (cf. ④ in Figure 5). Thereby, attributes can be hyperlinks referencing (cf. *S4* in Section 3.2) other concepts or plaintext. As illustrated, these attributes can also be used to reference a concept of a certain type (cf. ⑤ in Figure 5), in this case a person or group.

Specifying additional attributes changes the model behind a concept immediately. On the basis of existing and already structured data, a model emerges. Based on this model and previously stored data, recommendations for attributes can be made to encourage users to use this particular attribute (cf. ⑥ in Figure 5). The platform also empowers stakeholders to leave comments on the page (cf. ⑦ in Figure 5) for discussing issues they have with a certain service description, a used function, etc. These comments are then represented with the additional information of who actually wrote the comment (cf. ⑧ in Figure 5), so that a discussion about an issue in the description can be started instantly (cf. *S5* in Section 3.2). Any changes to a page are stored in a repository, i.e. versions are captured and are easily accessible (cf. ⑨ in Figure 5). For each version, it is marked *who* changed *what*, *when*. Finally, the last editor is shown on the page (cf. ⑩ in Figure 5) so that the Service Engineer sees *who* edited the service last.

[Image „Figure5-service.tif“ here]

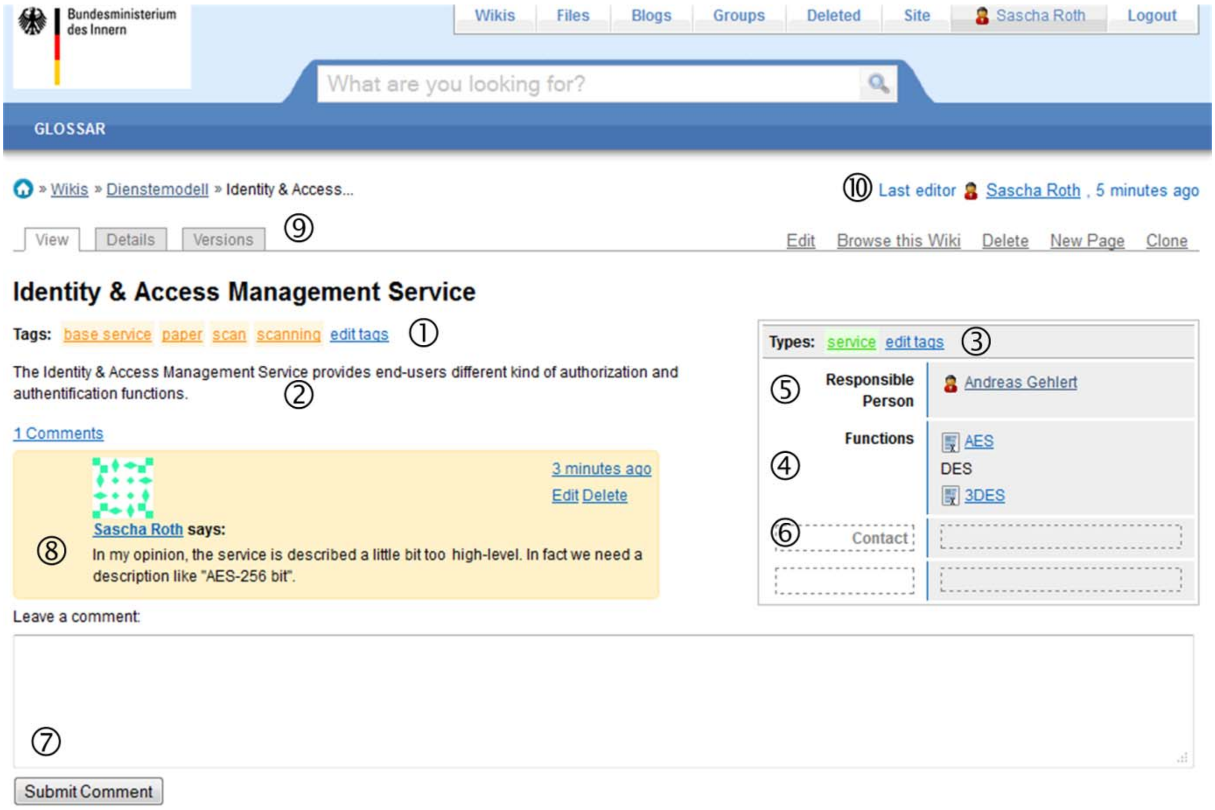

Figure 5: Service description example using Hybrid-Wiki concepts

Finally, the Service Engineer is concerned with *aligning the service with the service catalogue* (cf. step ❺ in Figure 2). Therefore, the Service Engineer compares the identified services with the existing service catalogue in more detail.

Using aforementioned full-text descriptions and attribute-value pairs (cf. *S1* in Section 3.2) enables the Service Engineer to employ another Enterprise 2.0 service artifact, i.e. using a full-text and faceted searches (cf. *S3* in Section 3.2). Thereby, full-text searches (cf. ① in Figure 6) can be combined with faceted searches. In this example, the facet selected refers to the concept of function (cf. ② in Figure 6), so that the functions are filtered according a certain attribute value (cf. ③ in Figure 6). The *auto-completion* feature will suggest only attributes that are relevant for the selected concept.

Simultaneously, Solutions Owners can help to contribute their knowledge to complete the service description (cf. step ③ in Figure 2). For instance, they could have objections against the usage of DES encryption. In turn, Solution Owners are notified if changes to subscribed services occur, i.e. if the Service Engineer removes DES encryption from the service's description.

Employing such techniques, the Service Engineer finds that the “Paper Document Handling” service also supports E-Mail that can be used to notify the bidders once their bid is received. Reusing this function comes with the advantage, that an existing IT solution can also be reused and that the effort to realize the "electronic procurement" service can be reduced. By a comment of a Solution Owner, the Service Engineer finds out that there is a customer relationship management system in use, which provides a service with the same name. A detailed analysis of this service reveals that it contains a function "Contract Management", which could support the activity "Buy Goods or Services" partially. In this way, the Service Engineer contributes to the business by automating activities of the business process, which were not selected for automation by the business. Figure 7 represents the result of this analysis. All elements are highlighted, which are modified in the final step of the method.

[Image „Figure6-search.tif“ here]

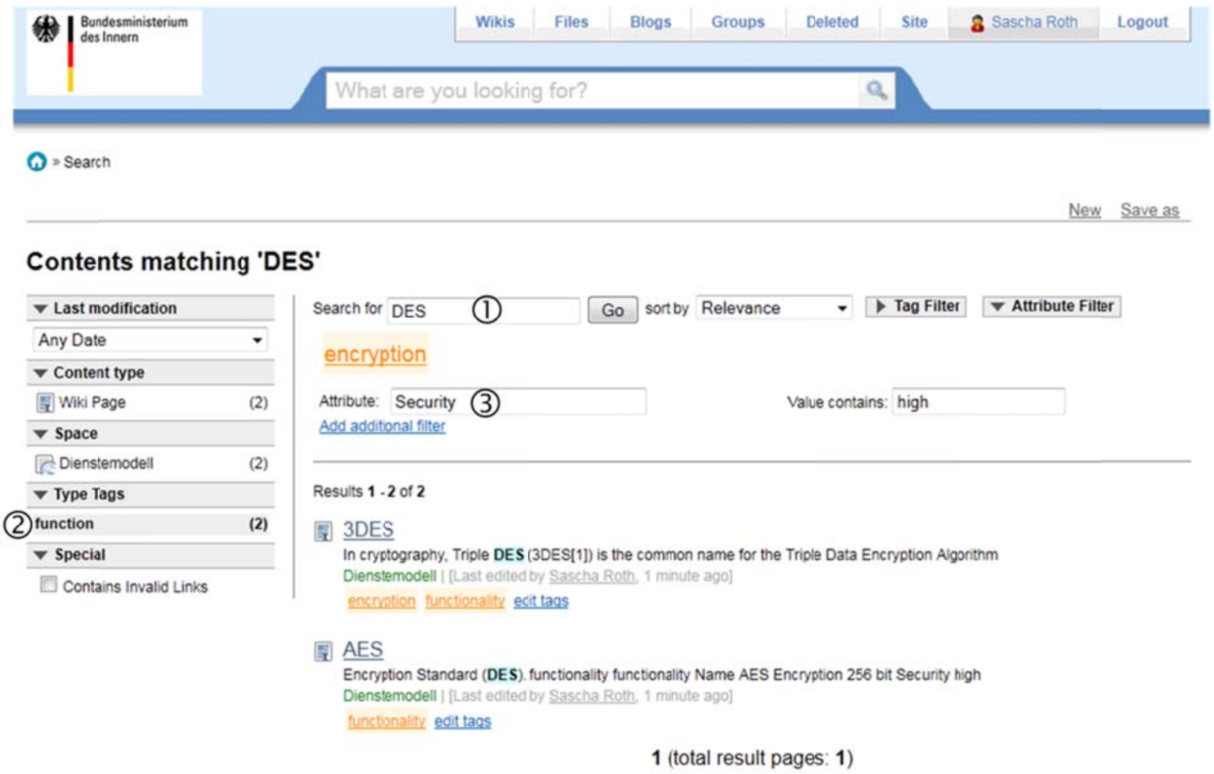

Figure 6: Full-text search combined with an attribute filter

[Image „Figure7-Metamodell-eGovPaper-2-3.tif“ here]

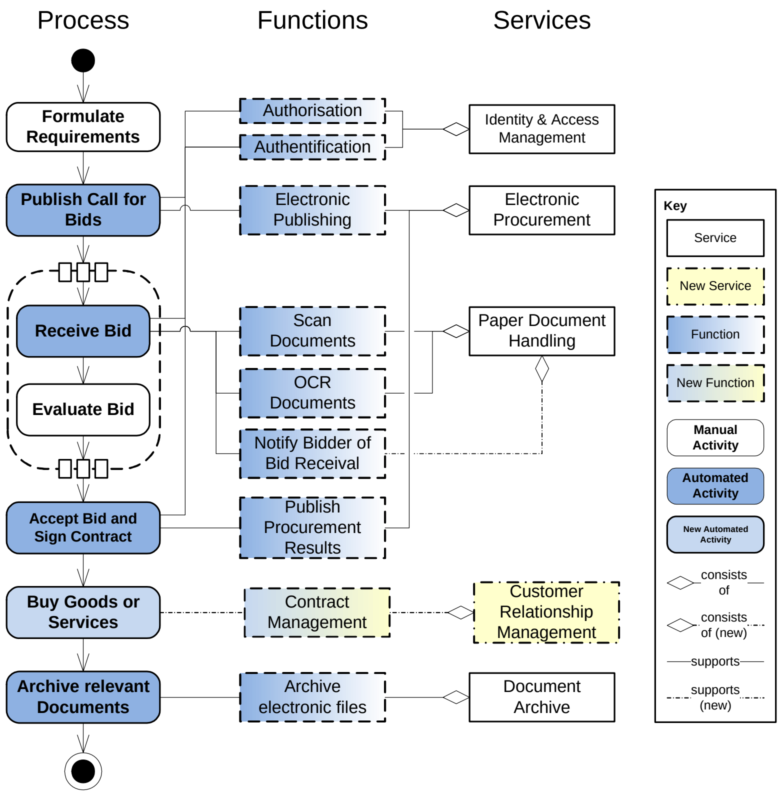

Figure 7: Process, functions, and services after comparison with the Service Catalogue

6 FUTURE DIRECTIONS

The introduced method to derive services from business processes contributes to a closure of research gaps discussed before (cf. Section 3). In this section, we discuss the underlying assumptions (A1-A3) to draw conclusions about the generalizability of the method. This discussion also provides insights into future research.

According to assumption A1 (cf. Section 2, p. 3), the method cannot rely on a single business process modeling technique. On the one hand, this ensures its applicability in many different projects with varying prerequisites. Additionally, it fits well to the situation found in practice, where modeling techniques are commonly adapted instead of being used by the book. However, the core set of elements (e.g. activities, control flow) is the same for most modeling notations so that the method is suitable for most process modeling notations. On the other hand, each modeling notation also has specific characteristics, such as e.g. different types of actions or events. A method which utilizes such characteristics during the derivation of services might, therefore, be able to achieve better results, but it is then also fully dependent on a specific language. Whether this might be an advantage or in fact turn out as a drawback in practice remains to be examined more closely.

According to assumption A2 (cf. Section 4.1, p. 8), the presented method only uses activities to derive services from business process models. As a consequence, it disregards the analysis of data structures, which generally form an important aspect during the modularization of systems (Parnas, 1972). In practice, however, business process modeling activities oftentimes do not encompass the documentation of processed data items. Our method thus is applicable even in such scenarios where many other proposed approaches cannot be used anymore. We will need to examine whether the proposed method can be extended in a way that it can also analyze data structures when present in business process models.

Assumption A2 also implies that a business process model is required before the proposed method can be used. This means that the proposed method cannot directly be applied to unstructured or ill-structured processes, which can usually not be expressed as process models. An interesting approach to tackle this problem would be to allow hybrid wiki pages as inputs of the method. As wiki pages can be generally used to describe unstructured data and, therefore, also unstructured processes, it is sensible to assume that the hybrid wiki approach can be used to describe unstructured and ill-structured business processes. Nevertheless, the proposed method needs then to be extended to accept a business process description from a wiki page as input. The evaluation of this idea is subject to future research.

Assumption A3 (cf. Section 4.4) requires that implicit functions can be identified by the Service and Process Engineers. This is an essential requirement, as different processes include different implicit functions which have to be incorporated in services. However, no general rule for a systematic identification of implicit functions can be stated. Hence, we will need to further examine the application of the proposed method in order to determine how reliable the identification of implicit functions is in practice. Additionally, suggestions of common implicit functions should be provided to support the service and process engineers.

Furthermore, it has to be evaluated in how far the current method that has been created for the German Federal Government can be seamlessly applied to equal problems in industry. Hence, the question whether all underlying assumptions are endurable in industrial practice as well has to be examined. Another key factor for its applicability is the acceptance of the method by all involved stakeholders. We therefore intend to further examine the users' commitment to the method once it has been applied in a significant number of projects. So far, the method proposed in this chapter has repeatedly produced usable services from a business perspective. We will have to examine the resulting service landscape with respect to design characteristics such as coupling and cohesion measures, though.

While the presented approach offers a high degree of flexibility, it also could lead to conflicts, e.g. disagreements on certain attribute names. The presented approach could also lead to a 'huge model', i.e. a tremendous level of detail and detail imbalances in the model could occur since people differ in the way they abstract. Merging and gardening mechanisms have to be developed and applied so that unnecessary details can be abstracted. The application of Enterprise 2.0 techniques often relies on single entities, i.e. a driver. Put in other words, it is crucial that such a driver is within the project, since emergent data modeling relies on contributors to reach a critical mass. It is yet to prove, whether or not emergent data models reach higher user acceptance rates than traditional models.

Moreover it will be interesting to see if the introduced approach scales in the context of government especially with a large number of contributing stakeholders which are widely scattered across several departments. However, we are confident that this will be rather uncritical since Web 2.0 technologies successfully apply in many different large-scale projects, e.g., in Bachmann and Merson (2005), the authors describe their experiences made with a Wiki used to create architecture documentation in a collaborative environment.

7 CONCLUSIONS

In this chapter, we proposed a method to derive services from business processes. The method consists of a data model describing the structure of the method's results; a role model describing the competencies needed to execute the method; and a process model describing the required activities and their sequence to produce the results of the method. The proposed method takes business processes as inputs and produces services as outputs.

Due to the specific organizational requirements of the public sector, the method needs to be generic regarding its inputs. Therefore, we could not presume a specific process modeling technique. In addition, we could not assume that the method has a fixed data model. Therefore, the method includes Enterprise 2.0 techniques, wikis, and an *open templating* mechanism, which link different data sources in socio-technical systems.

We see future work in the following fields:

- *Evaluation*: The method should be evaluated in a project applying a case study or action research method. We expect valuable insights from such an evaluation regarding the suitability, understandability and usefulness of the proposed method. It would especially be interesting to examine whether the connection between process modeling, service engineering and Web 2.0 techniques works as expected in practice.
- *Extension*: The method should be prepared for more specific inputs. Especially, process modeling became very popular in the public administration over the last five years. Once the process modeling technique matures in public administrations, more advanced modeling techniques are being used. Therefore, it is very likely that the additional information codified in those process models lead to better services. In addition, there may be a potential to automate the process of deriving services from business processes.
- *Guidelines*: When applying the method in different organizational units, e.g. in different public administrations, it is very likely that modelers of those units apply different conceptualizations to produce the models. This may lead to incompatible models. Therefore, the method should be extended by guidelines to achieve a higher degree of consistency.

8 REFERENCES

Aier, S., Gleichauf, B., & Winter, R. (2011). *Understanding Enterprise Architecture Management Design - An Empirical Analysis.* Paper presented at the 10th International Conference on Wirtschaftsinformatik, Zurich.

Aier, S., & Winter, R. (2009). Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. *Business & Information Systems Engineering, 1*(2), 150-163.

Azevedo, L. G., Santoro, F., Baião, F., Souza, J., Revoredo, K., Pereira, V., et al. (2009). *A Method for Service Identification from Business Process Models in a SOA Approach.* Paper presented at the Enterprise, Business-Process and Information Systems Modeling, Amsterdam.

Bachmann, F., & Merson, P. (2005). *Experience Using the Web-Based Tool Wiki for Architecture Documentation*: Software Engineering Institute, Carnegie Mellon University.

Birkmeier, D. Q., Klöckner, S., & Overhage, S. (2009). A Survey of Service Identification Approaches: Classification Framework, State of the Art, and Comparison. *Enterprise Modelling and Information Systems Architectures, 4*(2), 20-36.

Brinkkemper, S., Saeki, M., & Harmsen, F. (1999). Meta-Modelling Based Assembly Techniques for Situational Method Engineering. *Information Systems, 24*(3), 209-228.

Buckl, S., Marliani, R., Matthes, F., & Schweda, C. M. (2011). *Dynamic Virtual Enterprises - The Challenges of the Utility Industry.* Paper presented at the International IFIP WG5.8 Working Conference on Enterprise Interoperability, Stockholm.

Büchner, T., Matthes, F., & Neubert, C. (2009). *A concept and service based analysis of commercial and open.* Paper presented at the International Conference on Knowledge Management and Information Sharing, Madeira.

Chang, S. (2006). *Are they willing to contribute? Prosumer characteristics among the Australian youth.* Paper presented at the Digital Natives in Australia and Korea, Conference at the University of Melbourne.

Dietz, G., Juhrisch, M., & Grossmann, K. (2011). *Inherence of Ratios for Service Identification and Evaluation.* Paper presented at the 17th Americas Conference on Information Systems, Detroit.

Erl, T. (2005). *Service-Oriented Architecture - Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall.

Erradi, A., Anand, S., & Kulkarni, N. (2006). *SOAF: An Architectural Framework for Service Definition and Realization.* Paper presented at the IEEE International Conference on Services Computing 2006.

European Commission (2010a). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services': Annex I - European Interoperability Strategy (EIS) for European public services*.

European Commission (2010b). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services': Annex II - European Interoperability Framework (EIF) for European public services*.

European Commission (2010c). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions: A Digital Agenda for Europe*.

Frederiksa, P. J. M., & van der Weideb, T. P. (2006). Information Modeling: The Process and the Required Competencies of its Participants. *Data & Knowledge Engineering, 58*(1), 4-20.

Gehlert, A., Schermann, M., Pohl, K., & Krcmar, H. (2009). *Towards a Research Method for Theory-Driven Design Research.* Paper presented at the 9. International Conference on Wirtschaftsinformatik, Wien.

Golder, S. A., & Huberman, B. A. (2005). The Structure of Collaborative Tagging Systems Retrieved May 17, 2011, from http://arxiv.org/abs/cs.DL/0508082

Grundgesetz für die Bundesrepublik Deutschland 65 (1949).

ISO/IEC (2007). *Enterprise integration - Constructs for enterprise modelling* (No. ISO/IEC Standard 19440): International Organization for Standardization.

IT Planning Council (2010). *National E-Government Strategy*.

Jain, H., Chalimeda, N., Ivaturi, N., & Reddy, B. (2001). *Business Component Identification - A Formal Approach.* Paper presented at the 5th IEEE International Conference on Enterprise Distributed Object Computing, Seattle.

Klose, K., Knackstedt, R., & Beverungen, D. (2007). *A Stakeholder-based Approach to SOA Development and its Application in the Area of Production Planning.* Paper presented at the European Conference on Information Systems, St. Gallen.

Kohlborn, T., Korthaus, A., Chan, T., & Rosemann, M. (2009). Identification and Analysis of Business and Software Services - A Consolidated Approach. *IEEE Transactions on Services Computing, 2*(1), 50-64.

Kurpjuweit, S., & Winter, R. (2007). *Viewpoint-based Meta Model Engineering.* Paper presented at the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures, St. Goar.

Leuf, B. C. (2001). *The Wiki Way. Quick Collaboration on the Web.* Reading, MA: Addison-Wesley.

Matthes, F., Neubert, C., & Steinhoff, A. (2011). *Hybrid Wikis: Empowering Users to Collaboratively Structure Information.* Paper presented at the 6th International Conference on Software and Data Technologies, Seville.

McAfee, A. P. (2005). Enterprise 2.0: The Dawn of Emergent Collaboration. *MIT Sloan Management Review, 47*, 21-28.

Ministers of the European Union (2009). *Ministerial Declaration on eGovernment*.

O'Reilly, T. (2008). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*: O'Reilly Media.

Parnas, D. L. (1972). On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM, 15*(12), 1053-1058.

Patig, S., & Casanova-Brito, V. (2011). *Requirements of Process Modeling Languages – Results from an Empirical Investigation*. Paper presented at the 10th International Conference on Wirtschaftsinformatik, Zurich.

SAP (2005). *Enterprise Services Architecture: Enterprise Services Design Guide*: SAP AG.

Schermann, M., Gehlert, A., Krcmar, H., & Pohl, K. (2009). *Justifying Design Decisions with Theory-Based Design Principles*. Paper presented at the 17th European Conference on Information Systems, Verona.

Surowiecki, J. (2004). *The Wisdom of Crowds*. New York: Random House.

Wagter, R., van den Berg, M., Luijpers, J., & van Steenbergen, M. (2005). *Dynamic Enterprise Architecture: How to Make IT Work*: John Wiley.

Winkler, V. (2007). Identifikation und Gestaltung von Services - Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich. *Wirtschaftsinformatik, 49*(4), 257-266.

9 ADDITIONAL READINGS

The readings are organized according to the main lines of argumentation of the chapter: Firstly, readings explaining the political setting in Europe and Germany are listed (Political Background). The second reading list contains techniques to identify services and the third list includes relevant readings on the topics Web 2.0 and Enterprise 2.0.

This section is aimed to give the interested reader an overview of the before-mentioned topics. Therefore, we restricted ourselves to overview papers and/or books while more detailed papers on the topics are omitted.

9.1 Political Background

European Commission. (2010a). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services': Annex I - European Interoperability Strategy (EIS) for European public services*.

European Commission. (2010b). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services': Annex II - European Interoperability Framework (EIF) for European public services*.

European Commission. (2010c). *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions: A Digital Agenda for Europe*.

Ministers of the European Union. (2009). *Ministerial Declaration on eGovernment*. Ministers responsible for eGovernment policy of the European Union (EU) Member States, the Candidate Countries and the European Free Trade Area (EFTA) Countries.

9.2 Techniques to Derive Services

Birkmeier, D. Q., Klöckner, S., & Overhage, S. (2009). A Survey of Service Identification Approaches: Classification Framework, State of the Art, and Comparison. *Enterprise Modelling and Information Systems Architectures, 4*(2), 20-36.

Erl, T. (2005). *Service-Oriented Architecture - Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall.

IBM Corporation. (1984). Business Systems Planning: Information Systems Planning Guide.: Internationl Business Machines Corporation.

Kohlborn, T., Korthaus, A., Chan, T., & Rosemann, M. (2009). Identification and Analysis of Business and Software Services - A Consolidated Approach. *IEEE Transactions on Services Computing, 2*(1), 50-64.

Parnas, D. L. (1972). On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM, 15*(12), 1053-1058.

SAP (2005). *Enterprise Services Architecture: Enterprise Services Design Guide*: SAP AG.

Szyperski, C., Gruntz, D., & Murer, S. (2002). *Component Software - Beyond Object-Oriented Programming* (Vol. 2). New York: ACM Press; Addison-Wesley.

9.3 Web 2.0 and Enterprise 2.0

Buckl, S., Matthes, F., Neubert, C., & Schweda, C.M. (2010) *A Lightweight Approach to Enterprise Architecture Modeling and Documentation*. Paper presented at the 22nd International Conference on Advanced Information Systems Engineering, Hammamet.

Bughin, J. (2008). The rise of enterprise 2.0. *Journal of Direct, Data and Digital Marketing Practice, 9*(3), 251.

Cook, N. (2008). *Enterprise 2.0: How Social Software Will Change the Future of Work*. Gower Publishing Ltd.

Drakos, N. (2007). *Magic quadrant for team collaboration and social software.* Gartner Research, ID Number: G00151493.

Edmunds, A. & Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. *International Journal of Information Management*, *20*(1), 17 - 28.

Koch, M. (2008). *CSCW and enterprise 2.0 - towards an integrated perspective*. Paper presented at the Bled eConference.

Matthes, F., Neubert C., & Steinhoff A. (2011). *Hybrid Wikis: Empowering Users to Collaboratively Structure Information.* Paper presented at the 6th International Conference on Software and Data Technologies, Seville.

McAfee, A. (2009). *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Harvard Business Press.

Stocker, A., & Tochtermann, K. (2009). *Exploring the value of enterprise wikis: A multiple-case study*. Paper presented at the International Conference on Knowledge Management and Information Sharing.

10 KEY TERMS & DEFINITIONS

This section serves as a glossary. Therefore, you will find all definitions also in the text.

Activity: An activity is the core element of a *process*. It describes one step of that process. An activity is also the basic element to derive functions.

Emergent Data Modeling: In contrast to traditional database systems following a schema first, data second approach, emergent data modeling follows the principle of data first, schema second. After collaboratively collecting data in a non-rigidly typed system, a schema emerges (over time with rising data) and can be extracted based on instance data.

Function: A function represents a functional requirement. Functions are always assigned to *activities*. A many-to-many relationship between functions and *activities* decouple the granularity of the *processes* from the granularity of the *services*. Functions result either from a functional requirement from *activities* which should be automated or a requirement which results from the IT support of an activity (implicit function). Examples of the latter case are login or encryption functions.

Process: A process is a set of *activities* connected by a control flow. The control flow describes the logical sequence of the *activities*. Please note that many process modeling techniques may include other elements such as organizational roles, data or events in processes. This additional information, however, is not needed for our method.

Process Engineer: The Process Engineer is responsible for designing business *processes*. S/he knows the processes of the organizational unit in focus and is able to express those processes in a modeling language, which distinguishes at least processes and activities (cf. Assumption A2). Additionally, the role includes the rights to (re-) design the business processes of the organizational unit and to decide, which *activities* should be supported by IT. Furthermore, the Process Engineer is knowledgeable about organizational aspects, which are not covered in the business process and which may lead to implicit *functions*.

Service: A service is a set of requirements, which is already supported by IT solutions or will be realized by IT solutions in the future. By IT solutions, we mean any software, or component thereof, which may realize a service. A service is made of *functions*.

Service Catalogue: The service catalogue contains all *services* of a public administration. It is the major means to foster reuse and to avoid the duplicate implementation of the *functions*.

Service Engineer: The Service Engineer is responsible for the *service catalogue*. S/he knows the structure and the content of the *services* in the service *catalogue*. In particular the Service Engineer should know which *services* are already realized by IT. Furthermore, the Service Engineer is empowered to decide, which *services* may be added to the *service catalogue* and has experience in identifying implicit *functions* and in identifying *services*. S/he has in-depth knowledge of the proposed method for deriving services from business processes.

Stakeholders: Stakeholders are represented by the different groups of professionals actually executing functions of a *process*. They have insights into the *process* steps, i.e. *activities* and corresponding *functions* from day-to-day application thereof in their professional occupations.

Solution Owners: The Solution Owners host implemented *service* realizations, e.g. a running business application offering electronic publishing services.