Service Registry Design: An Information Service Approach

Luís Ferreira Pires, University of Twente, the Netherlands

Arjen van Oostrum, University of Twente, the Netherlands

Fons Wijnhoven, University of Twente, the Netherlands and Wilhelms University of Muenster, Germany

ABSTRACT

A service registry is a Service-Oriented Architecture (SOA) component that keeps a 'catalogue' of available services. It stores service specifications so that these specifications can be found by potential users. Discussions on the design of service registries currently focus on technical issues, while service registries should take into consideration information needs of business domain users. In this regard, the authors consider service registries as information services and develop a comprehensive framework for designing service registries. This framework introduces aspects that determine a design space for service registries. In this design space, the authors identify views, requirements, processes, and means in the design of a service registry that supports the lifecycle information of a service. A vital part of these requirements is further implemented and demonstrated in a prototype built as a 'proof-of-concept' for the framework. This paper also discusses a case study used to evaluate the prototype. In this case study, a registry prototype has been populated with realistic services of a large insurance company, and 21 experienced IT and business professionals from a consultancy organization evaluated the prototype for its user satisfaction.

Keywords: *Design Science, Information Services, Service Discovery, Service-Oriented Architecture, Service Registry, SOA Governance*

INTRODUCTION

Service-Oriented Architecture (SOA) has been introduced with the promise that by offering functionality as services, business processes supported by these services can be more easily composed and executed (Papazoglou, 2008). However, a successful SOA implementation depends on various factors, like a structured decomposition of processes into services, appropriate management support, and *SOA governance* (Mahajan, 2006). SOA governance offers the measuring and steering capabilities that help organizations reach the objectives of their SOA implementations. A particular aspect of SOA governance involves service offerings, which should comply with organizational policies and norms. An overview of the services available in an organization should be in place, in order to avoid redundant offering of services with similar or identical functionality, or offering of services that are not relevant for any business process. Service lifecycle information is valuable for

DOI: 10.4018/jisss.2010100101

service developers, owners and providers and should be supported by the registry.

In SOA, a service registry is an architectural component that enables a service provider to publish service descriptions and enables a service consumer to find services based on their descriptions (Alonso, Casati, Kuno, & Vijay, 2004). Consequently, a service registry is like a 'Yellow Pages for services'. Service registry implementations based on standards like UDDI and ebXML mainly store technical information on services by means of a flat data model with limited search capabilities (Luo, Montrose, Kim, Khashnobish, & Kang, 2006). Service registries developed according to these standards are mainly suitable for runtime service users, which are interested in interface definitions and the technologies to reach the service at runtime. However, service specifications are also supposed to be used by enterprise architects, application developers and business process engineers, amongst others (Li et al., 2009). These service registry users often need other sorts of information than runtime users, such as the service goals or business value, which correspond to meta-information about the service (Ran, 2003). Furthermore, this information is meant to be understandable for (non-technical) human users, as opposed to the more technical information supported nowadays (Samavi, Yu, & Topaloglou, 2009).

The work reported in this paper has been motivated by the opportunities offered by service registries with respect to SOA governance once these registries are properly designed, and the need to improve the methods available to design these registries.

This paper provides a framework to design and implement service registries for storing, managing and disclosing service specifications. This requires a design that addresses both the technological and business means necessary to implement service registries that are more suitable to support SOA governance than the service registries available today. We consider a service registry as a special case of an information service, which transfers information goods from suppliers to consumers. We define a design space consisting of aspects (content, use features and revenue) and layers (design problem, business, process, infrastructure, prototyping and exploitation) that are relevant for developing effective information services (Wijnhoven & Kraaijenbrink, 2008).

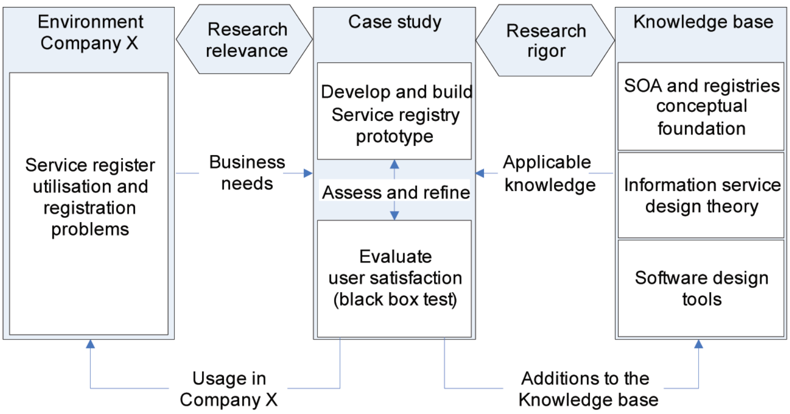
We consider service registries as information services in order to enhance the satisfaction of the consumers of service specifications. We test the validity of our framework by:

1. Assessing the feasibility of designing useful service regist~~ers~~ with this framework. Other approaches could be used for designing service regist~~ers~~, but we believe that by considering service registries as information services we can obtain registries that offer the highest value for their users. This is because the service design theory forces the designer to systematically include the actual business requirements in the service registry design and to consider the actual value of the information goods (service specifications) that a service registry is expected to deliver to its users.
2. Assessing the relevance of this approach by solving a concrete real life problem with a service registry in a case study. This is achieved by building a service registry prototype as a proof-of-concept that supports part of the aspects covered in our framework.
3. Performing a utility test in which registry users provide feedback by scoring their user satisfaction with the service registry prototype.

These steps correspond to the generate-test cycle ~~of~~ (Hevner, March, & Park, 2004), as shown in Figure 1.

Furthermore, we evaluate our results against design science guidelines of Hevner et al. (2004), such as artifact construction (we define a design methodology and build a concrete service registry), relevance (we address real-life problems) and utility (we test our results). We also contribute to design foundations

Figure 1. Our research approach in terms of design science. (Adapted from Hevner et al., 2004)


Environment Company X
Service register utilisation and registration problems
Research relevance
Business needs
Case study
Develop and build Service registry prototype
Assess and refine
Evaluate user satisfaction (black box test)
Research rigor
Applicable knowledge
Knowledge base
SOA and registries conceptual foundation
Information service design theory
Software design tools
Usage in Company X
Additions to the Knowledge base

by providing design constructs for service registries as information services.

This paper presents our design methodology, and illustrates ~~the~~ steps ~~of this methodology,~~ starting from the definition of business requirements and ending at the construction and evaluation of a service registry prototype in a case study. Therefore, the paper is further structured as follows: 'Design Methodology' describes our design methodology, by introducing a design space for service registries. 'Business Requirements' discusses business requirements from the stakeholders of a service registry. 'Process Requirements' discusses process requirements of interest for actors of the service registry processes. 'Infrastructure and Prototype Design' introduces the overall architecture of a service registry, and discusses the choices made in our prototype design and implementation. 'Exploitation and Evaluation' discusses the results of our case study, and 'Conclusion' gives conclusions and suggestions for future work.

DESIGN METHODOLOGY

Design Space

Three aspects should be considered in the design of an information service: the delivery of content to the user and all associated functions (e.g., content acquisition and aggregation), the delivery of use features for a better use and value experience of the delivered information good, a[illegible]e stream of revenues to be realized for the information service to be viable for the content owners and information service providers (Wijnhoven, 2008). In order to support these aspects, we analyze them at different layers. The top layer addresses 'why' the information service should be used ('Design problem and agenda'), the subsequent layers cover 'what' is needed to support the information service ('Business requirements' and 'Process requirements'), and the bottom layer addresses 'how' to design an infrastructure to support it ('Infrastructure'). The infrastructure layer does not only comprise the design of the IT infrastructure such as databases and networking, but also the structuring of employee tasks and tools to support these tasks. However, if we apply only these layers we may obtain a working information service, but for a successful and enduring system we should also consider the construction of a prototype ('Prototype construction') and its exploitation and evaluation for suitability ('Exploitation and evaluation') as explicit design layers.

Figure 2 depicts the resulting design space, which consists of the three aspects, supported by the six layers. Figure 2 shows that eighteen blocks are formed by the crossings of the six rows (layers) and three colored columns (aspects), which indicate the areas that have to be addressed when designing an information service.

Each design, for each type of information service and its instances, is likely to be different from others, due to its particular context. Therefore, it is difficult to prescribe any specific order in which the blocks of Figure 2 should be considered. Top-down, bottom-up and middle-out strategies can be used, as long as the results for each block remain aligned, i.e., they do not contradict each other and the results are mutually supportive.

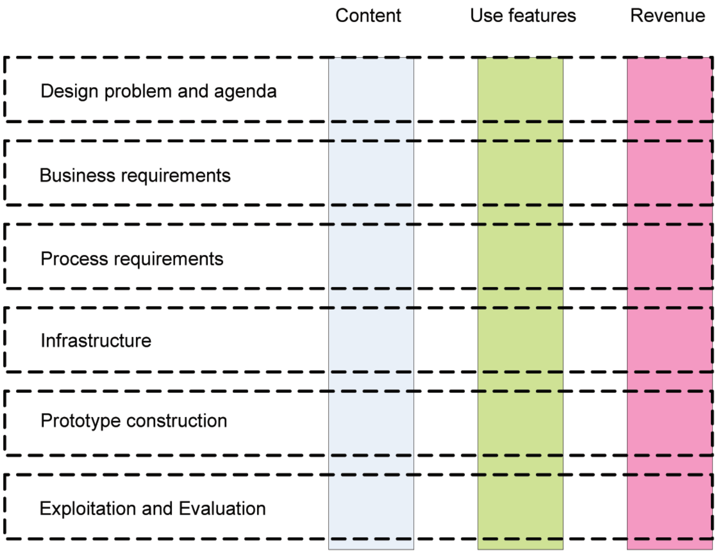
Content

The content of an information service is some information good, supplemented with some optional information (supportive information or meta-information). In general it is difficult to match these digital goods exactly with the user's demand, since the shear volume of variation options for digital information goods makes it hard to present the user with a limited number of choices. An information service has therefore the task of reducing the amount of possible information good variations to present meaningful and intelligible information to the user.

An information service may transform the information good in order to meet the user's request by delivering just what the user needs. It can do so by altering two dimensions of an information good, namely the Level of Representation (LoR, i.e., the percentage of coverage of all the data that can be delivered) and the Level of Conceptualization (LoC, i.e., the abstraction level of what is delivered, varying from elementary data to theoretical models). By increasing or decreasing either or both levels, the information good can be 'customized' to meet the user's needs.

Since we consider a service registry as an information service, the information good in this case consists of service specifications. The service specification in a service registry can be characterized as process information, in that it can be used to compose business processes. The use of accepted standards can help achiev~~ing~~ high quality, and is a prerequisite for exchanging process information (Van der Aalst & Kumar, 2003), which implies that service specifications should be described systematically, preferably

Figure 2. Design space for information services. (Adapted from Wijnhoven, 2008)



by using generally accepted rules or standards. Service specifications should contain both run-time and design-time information, with added supportive information. Not every specification detail is useful for every user, and therefore the LoR can be decreased to present the information that matches the needs of each specific user. For example, some users may only get the interface description as the result of a query. Additionally, the LoC can increase by giving users meta-information in addition to descriptions of how the service specification is used in other situations.

Use Features

Use features can be provided by the information service in order to increase value experience, from which user satisfaction is an important factor. Use features may consist of additional content interaction options, such as advanced search options and extra information about the content. An information service can become more attractive to potential users if it provides use features.

In case of competing service registries that provide the same information good, the support to use features can make a difference for the users to actually use a certain registry. In this paper, for the sake of simplicity we only consider service registries in a single organization or in a network of cooperating organizations. Use features like credibility or quality evaluations of former users can be helpful to decide whether or not to use an information service, but they are not so relevant when the users are obliged to use the service registries of their organization (or organization network), which is the situation considered in this work.

However, since we have aimed to design service registries with a high end-user satisfaction, some use features have been considered in our framework. Examples are help functions to formulate search queries, and the delivery of alternative results, possibly ranked according to their suitability.

Revenue

Although the digital nature of an information good makes distribution inexpensive, costs have to be made in order to create, maintain and provide the information good. In the case of a service registry, the quality of the service specifications has been identified as a key factor, which implies that the maintenance of the service specifications requires extra (human) attention. Revenue streams have to be created to cover the costs of both the service providers and of the organization supporting the service registry. In the case of service registries, information is transparent, in the sense that service specifications should be sampled by service providers, repeatedly used by registry consumers, and their value can be tested once they are acquired. Furthermore, service specifications are meant to generate social benefit in the organization or organization network. According to Womack (2002), since we expect many consumers and relatively few service registries, subsidy and a non-profit organizational unit is required to cope with revenue. However, the consumers still have to pay somehow for their use of the registry in order to create a revenue source to financially support the registry. Possible payment models can be based on subscriptions or on the actual use of the service registry. Payments can be enabled by the actual consumer or sponsor.

BUSINESS REQUIREMENTS

The service registry, as an information service, has several stakeholders. We identified four main stakeholders in an organization with respect to the service registry: *registry consumers*, *service providers*, *organization managers*, and *system and information maintainers*. These stakeholders should have a positive attitude towards the service registry, acknowledge its added-value, and have to be aligned with the goals of the service registry. Assuming these conditions hold, in the sequel we discuss the requirements of the service registry for the business layer for each stakeholder, and we present a value exchange model that considers the contents and revenue aspects of a service registry.

Consumer

Consumers are the end-users of the service registry. These may be people in an organization, or an application or software agent. These users have different demands for information, and therefore we distinguish between them. First, the registry can be used at design-time, where primarily meta-information (e.g., information about the service goals and intended usage) is requested. In this case, our design approach focuses on this (human) end-user. Second, users at runtime have a demand for primarily technical data. To capture both technical and business aspects of a service from the viewpoint of the different sorts of consumers, we adapt the seven specification levels for business components identified in Fettke and Loos (2003), as shown in Figure 3.

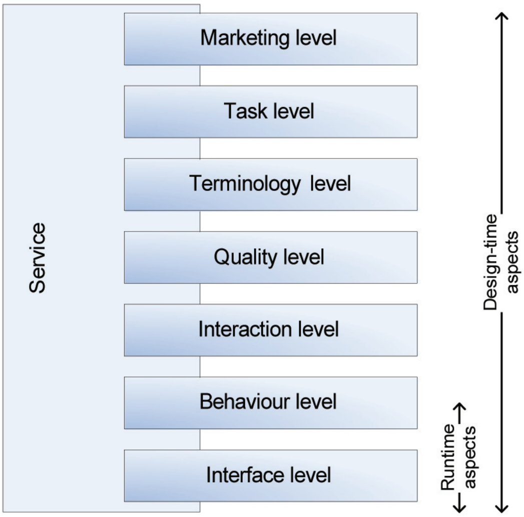
Figure 3 represents the information needs of the design-time service registry end-users, and since the interface level is included, also the needs of the runtime users are covered. These levels are the core information for design-time and runtime reuse of services. The terminology level describes concepts and elements that are used at all levels. The terminology level does not add any specific information to a service specification, but aims at avoiding ambiguity by defining concepts and terms in a structured and consistent way.

Provider

The service provider can be an organization, organizational unit, or any other entity providing some information good. Service registries provide service specifications as information goods, with the purpose of enabling the use of these services by others. Services have a lifecycle as any other IT system, and this lifecycle should be observed in order to keep the service useful and fit to the overall system architecture. Service providers have the responsibility of maintaining these services according to the service lifecycle.

Figure 4 depicts a service lifecycle in which the service lifecycle descriptions of Afshar (2007) and Larsen and Wilber (2005) are combined. Each lifecycle phase consists of various activities and to perform these activities the service stakeholders need information from earlier activities. For example, the testing

Figure 3. Levels of service specification. (Adapted from Fettke & Loos, 2003)



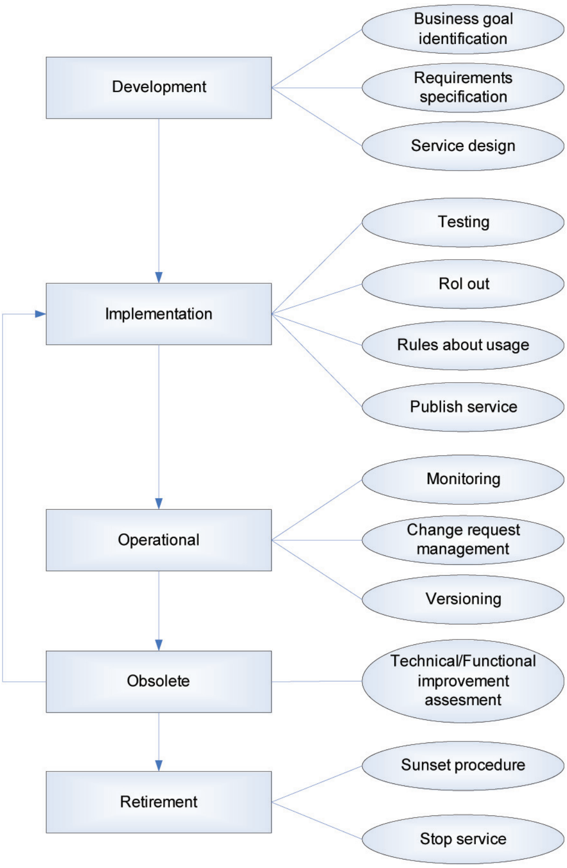
activity needs information that has been stored in the service design phase.

A service registry can hold information on the status of the services with respect to their lifecycle. A service specification can be published in the registry already at the beginning of the development phase and can record all relevant information for this and the subsequent phases. Lifecycle information at this point should capture the purpose for the development of this service. The stakeholders involved in the service lifecycle can use the service registry to retrieve information, supporting in this way their different information needs. Changes in business needs or technical developments may cause changes in the service (Bose & Suumaran, 2006). During its lifecycle, a service may change quite often, which may improve the value of the service for its users. Before being changed, a service may be evaluated to determine whether the change effort is justifiable. The service registry can support the service lifecycle by providing information not only about the status of a service, but also its change requests or historical change data (Belter, 2008). Service lifecycle management requires supportive information that is directly linked to a specific service, but does not contain any usage information.

Management

The management of an organization is responsible for positioning the service registry in the organization. Management can also take

Figure 4. Service lifecycle



the role of a financial sponsor of the service registry, for example, when the organization elicits and charges all costs through an (internal) cost model. Analogously to application governance in an application portfolio, services can be governed in a service portfolio, allowing managers to measure and steer them. The knowledge on available and popular services in an organization gives the ability to minimize the costs of inefficient services usage, caused by having redundant services and services that are not used at all (Yu, Liu, Bouguettaya, & Medjahed, 2008).

Ideally, all services that are bought or developed should comply with standards, so that their integration in the overall architecture can be facilitated (Bose & Suumaran, 2006). Services should all be published, so that they can be readily found and made available for use. They should be effectively used, so that no unused services are maintained, they should have no redundant functions, so that no extra costs are made to support duplicated functionality, and they should be used under optimal contracts, for example, by offering discounts on the costs per invocation to frequent users. The service registry can be the central place where the information necessary to steer the IT support is stored. This information consists of contracts, service status, service consumers and standards, to name just a few (Li et al., 2009). Along with organizational policies, the readily availability of this information is essential to optimize the service portfolio and minimize costs. The portfolio information related to a service specification is considered to be supportive information (Winkler, Cardoso, & Scheithauer, 2008).

Furthermore, management is normally also interested in information about the use of the service registry itself, especially in case the users of the service registry have to pay for accessing its functionality. This usage information is not directly linked to a certain service specification and is called coordination information. It typically consists of the number of users and the costs related to the use of the service registry.

Maintenance

Maintenance can be divided in maintenance of the service registry as an information system, and maintenance of the service specifications stored in the registry. The technical contents of the service specifications are the responsibility of the service provider, so that here we only consider generic mechanisms for enhancing the quality of these specifications.

Service registry system maintenance concerns processes for functional, application and exploitation management of the service registry software system. The organization in which the service registry is implemented should have proper structures to perform these maintenance tasks. The service registry can be designed in a way that facilitates this work by providing relevant logs and other information about the system operation depending on the requirements set by the organization (Gorbenko, Romanovsky, & Kharchenko, 2008). This information is not directly related to a service specification, and therefore should not be considered as supportive information.

Maintenance of service specifications is a key factor for a successful service registry. If service specifications are not properly maintained, the consumers cannot rely on these specifications and end up distrusting the services (Al-Masri & Mahmoud, 2008). Maintenance of service specifications concerns mainly the quality of these specifications, i.e., how faithfully the actual service is represented by the service specification, which is the responsibility of the service provider. High-quality of the service specification should ensure that the consumer has correct expectations of the service, so quality is essential in the operational phase in the service lifecycle. Maintenance personnel concentrate their efforts on improving the supportive information relevant for the consumer and provider views. The effectiveness of the service discoveries should be monitored and analyzed in order to improve the matching of service requests to service specifications, and ultimately the services they represent. By assessing the way users search for information, the service specification maintenance can improve the discovery of service specifications.

The terminology level presented in Figure 3 describes terms of the service specification as used at the other levels. These terms can be combined and used to create a domain ontology for service discovery. This ontology can be used for improving search and determining the relevance of a search query result (Verma et al., 2005).

Value Exchange Model

At the business layer we can define the values that are exchanged between stakeholders through the service registry in order to satisfy these stakeholders. The intended interactions with the service registry have been considered to generate requirements for the service registry at this layer. User satisfaction should be enough for the service registry to be used even without obligatory regulations in the organization. A value exchange can refer to the content, use feature, or revenue aspects, and it specifies which values are delivered by a stakeholder in return to other values. The specific realization of these exchanges is ignored at this layer, and is considered at the process layer. Value exchanges between the stakeholders can be modeled, for example, by the e3value technique of Gordijn and Akkermans (2001). In addition to the stakeholders who are directly involved in the exchange of content, the revenue aspect imposes that some form of sponsorship is needed, so that it should also be modeled.

Different alternatives can be chosen when defining value exchanges. For instance, we may have different choices for organizing sponsorship. Figure 5 shows one possible e3value model for the service registry, which consists of a high-level scenario in which the sponsor (possibly the management of the organization) directly subsidizes the service registry. In a more detailed model, the value exchanges can be further specified, and internal value exchanges can also be modeled.

Figure 5 shows four value exchanges supported by the service registry: (i) the service provider provides a service specification to the service registry, which should pay for the specification in return; (ii) a runtime user can initiate a value exchange in which (some part of) a technical specification is obtained in exchange for a specification fee; (iii) an end-user can pay a specification fee and gets several value objects in return, and (iv) a sponsor can pay for (part of) the specification fee, for instance, in return for feedback and visibility, or to increase the chance that the registry is used by consumers in case these customers are reluctant to use the registry. The service registry offers various use features and search assistance to help the end-user retrieve the (design) service specification. The service registry then delivers a specification, which is the core value of the service registry, i.e., the task of the service registry is to deliver the specification as efficiently as possible. To deliver the specification and meet the expectations of its users, the specification should be complete, of high quality and faithfully represent the actual service.

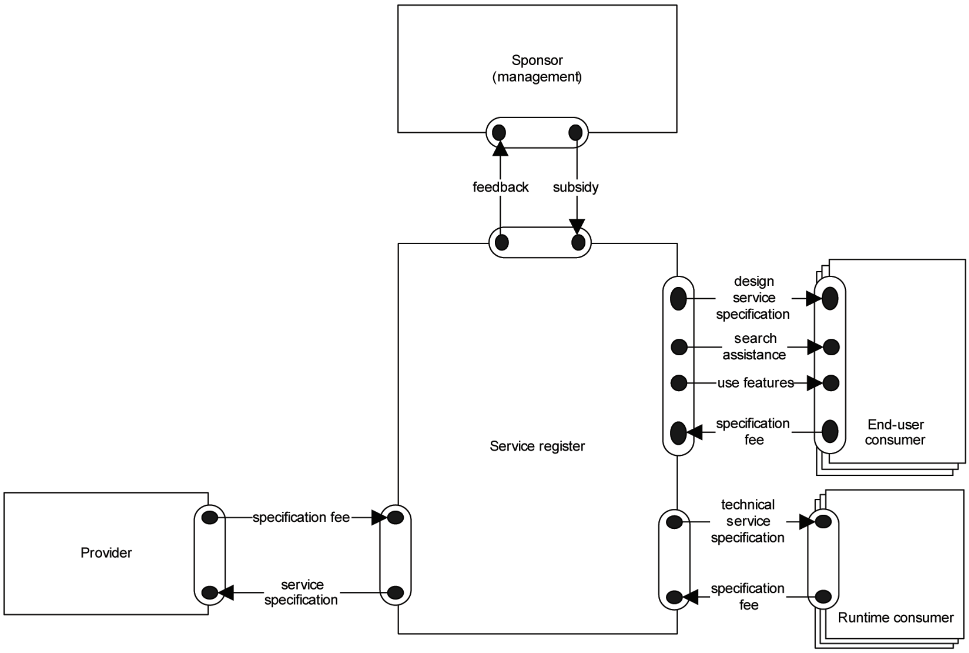
PROCESS REQUIREMENTS

At the process layer, we define the activities for delivering content, facilitating content usage, and handling usage costs and revenues, in contrast with the business layer which ignores all information service internal activities. Requirements for the process layer activities are discussed below by focusing on the interactions with the end-users.

Content Delivery

A service registry delivers service specifications to consumers. In order to perform this task, the service registry aggregates and stores the available service specifications. In order to process a request, the service registry should not perform any interactions with the service provider or other actors. Therefore, the service registry should have stored all the content necessary to answer a request of an end-user before the request is issued. The delivery of results can be customized to meet the request

Figure 5. High-level e3value model for a service registry



of a human end-user, for instance, by altering the level of representation (LoR). In this case, the service registry does not deliver the entire available service specification, but makes a selection of the available information for the end-user instead. When possible, it can also alter the level of conceptualization (LoC) by delivering fewer details and giving information at a higher abstraction level, like application domain information.

During publication, the information good content (service specification) is delivered by the service provider to the service registry. The end-users demand a high quality of the service specification to achieve a high end-user satisfaction, which is determined by how faithful the service specification represents the actual service. Therefore, before the service specification (or an updated version of this specification) is added to the content base, a series of activities have to be performed on the service specification to ensure that the content meets the quality standards set by the service registry.

Use Facilitation

Along with the content, use features are delivered to positively influence the factors that lead to a high end-user satisfaction. Factors that lead to end-user satisfaction are content, accuracy, format, ease of use and timeliness (Doll & Torkzadeh, 1988). The following requirements can be formulated with respect to these factors:

- Content: the service registry has to deliver the precise information needed by the users. Therefore, the needs of the users have to be determined, e.g., by helping the users formulate their information needs.
- Accuracy: accurate delivery consists of presenting the best matching service specification(s) to the users. The service registry can increase the accuracy by

providing alternatives to the users. Using ontologies, the service registry could determine the relevance of the presented results and suggest other specifications which are close enough to the presented ones.

- Format: the presented results should be shown in a clear and useful format. The service registry can assist in this by increasing usability, e.g., by providing different formats so the user can choose the appropriate one. A possibility is to offer the same information in different formats, e.g., in documents using formats like HTML, XML, PDF, MS Excel or MS Word.
- Ease of use: ease of use is increased by providing user-friendly interfaces with help functions and the use of standard information presentation.
- Timeliness: timeliness is influenced by different issues. The technical issue determines the speed at which a request is processed and a response is delivered by the service registry system. The delivery speed is subject to the available processing power, and speed of the computing and networking system. Another issue is the freshness of the service specifications, i.e., whether specifications are still valid or are outdated. While the technical issue can be influenced in the development and deployment of the service registry system, information freshness can be preserved by applying proper maintenance procedures. For example, the service registry system can assign a time-to-live to service specifications, and when the time-to-live of a service specification expires, maintenance personnel can be asked to check whether the specification is still valid or should be updated or removed.

By addressing some of these factors simultaneously when the content is delivered we expect that the experienced value of the content can be enriched.

Handling of Usage Costs and Revenues

The way subsidy is granted, i.e., which revenue model is chosen, determines how the usage costs can be handled. The billing strategies to be used can be a flat-fee subscription or pay-per-unit. Other approaches to cover usage costs (e.g., advertisement, syndication, additional merchandise, pay-per-data-packet) are based upon an open market for content delivery, which is not applicable to most SOA contexts and the setting we consider in this paper. Functional requirements to handle usage costs can be to check the subscription status of the end-users before granting access to the system.

Process Model

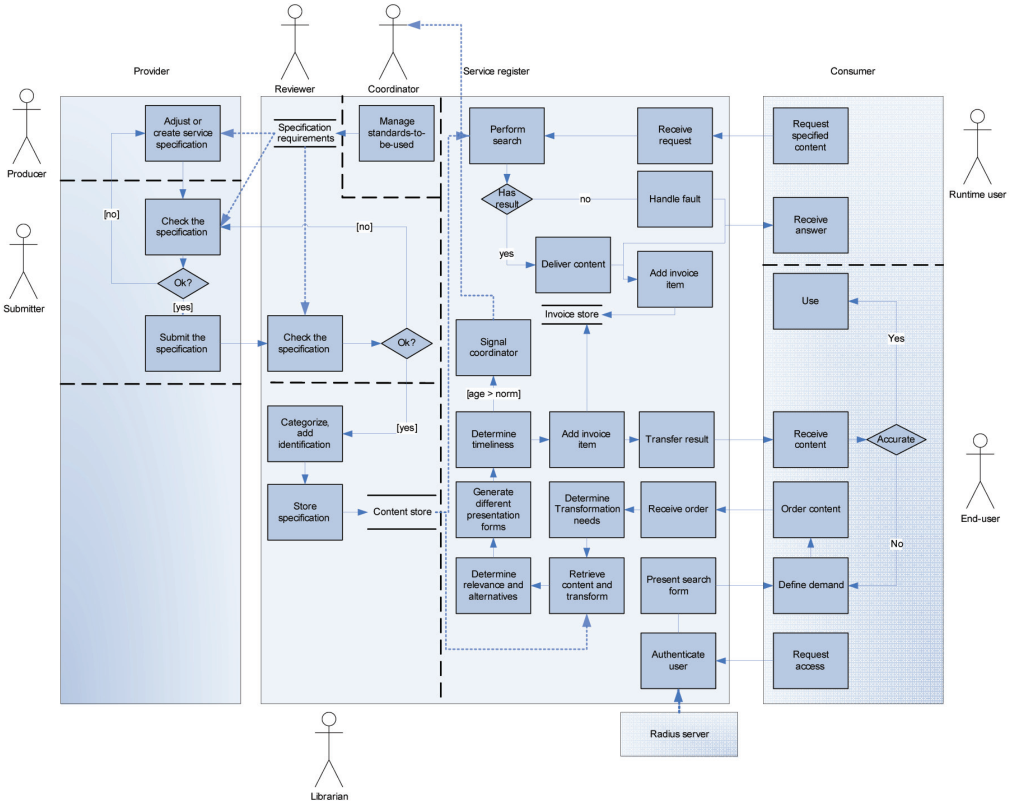
Figure 6 shows a process model that incorporates the process requirements described above. It assumes that the maintenance of a service specification is part of the service registry unit.

The actor roles that can be played by the stakeholders and that are relevant in this process have been drawn on the borders of Figure 6. The part of the model that corresponds to each role is indicated with striped lines. Dotted lines indicate a stream of information, while a solid line depicts a process step, possibly containing information elements. More detailed process steps can be drawn by zooming in on process elements. This process model only shows the delivery of content by providers and to consumers. Other processes (involving other stakeholders) could be represented in a similar way.

INFRASTRUCTURE AND PROTOTYPE DESIGN

Figure 2 shows the infrastructure layer immediately below the process requirements layer. The infrastructure layer deals with the mechanisms necessary to support the processes. Since we aim at creating an effective design for a service registry, we consider that a design is effective when it satisfies the end-user in his/her informa-

Figure 6. Process activities chart



tion needs. For this we used the five factors of Doll and Torkzadeh (1988) (content, accuracy, format, ease of use and timeliness) as our main design focus. Below we discuss the means to realize the processes related to the service registry, we introduce our overall architecture, and we present our prototype.

Realization Means

The processes we identified for the service registry (content delivery, use facilitation and handling of usage costs) can be realized using information, technical and organizational means. These means support these processes, and can be identified by systematically investigating what is necessary for each of these processes.

Processes need information to perform tasks for the service registry. Information means consist of the available and accessible data, and mechanisms to allow data to be provided, i.e., the organization of this data and resources, and the management of data freshness (Wijnhoven & Kraaijenbrink, 2008). Technological means are necessary to facilitate the activities of the service registry. Organizational means consist of lists of responsibilities, assignment of tasks and coordination principles. Furthermore, an organization unit can be created to carry the responsibility for the service registry. Table 1 identifies information, technological and organizational means for each main process, without trying to be comprehensive.

Prototype Construction

In order to evaluate the users' satisfaction with a service registry developed according to

Table 1. Information, technological and organizational means for the main processes

Process	Information means	Technological means	Organizational means
Content delivery	- List of available service specifications - Requirements for new services - Requirements for standards to use when creating a service specification - Checklists for checking the validity of service specification - Consumers requests - Consumers authentication - List of pending specifications - Information about information consumers needs - Structured data storage and retrieval - Contracts with providers - List of providers - Knowledge about consumer usage - Data models for efficient storage and retrieval	- Tool for syntax checking of service specification - Transformation tool for level of representation - Querying software - Interfaces for receiving content - Communication channel with providers and consumers - Databases to store specifications, requirements and invoices	- Service registry system programmers - Service registry system maintenance people - ICT services management to support client interactions - Providers network - End-user support - Coordinator to create/manage specification requirements - Procedures to add service specifications - Archiving procedures - Librarian - Database administration
Use facilitation	- Data about consumers needs - Supportive information database - Data models for retrieval - Mapping of consumers needs onto available data - Ontology data and relations - Relevance data - Modification date of service specification - Data about preferred user interfaces and possibilities - Links to supportive information (help functions) - Direct links between service specifications and supportive data	- Interfaces - Interactive search determination tool - Search system - Ontology searching mechanism - Data communication network - Presentation/customization tool - Database with feature information - Hard/software platform	- End-user support unit - Librarians to manage supportive information - Librarians to analyze system's usage to improve matching - Provide search and results options to clients - Create and maintain ontological links - Client feedback collection procedure
Costs and revenue management	- Contracts with consumers - Contracts with providers - Data delivery - Data reception - Subscription data	- Billing system - Subscription system - Usage logging system	- Providers ('account management') - Consumers - Sales administration - Contract management

our framework; we developed a prototype by focusing on the functional requirements of the end-users, namely the main use case of 'searching for service specifications'. Performance and availability requirements are crucial for a system to be used in production, however, are less relevant when limited functionality is demonstrated in a prototype. The main goal of a service registry is to provide service specifications to human end-users. To be able to provide the specification to the user, the system first needs to know what the user is looking for. From interview sessions with potential service registry users we have learned that the system should be able to search for specifications in different ways, for example, based on the service name or the function provided. The prototype should also assist the user with the formulation of the search question.

We built a prototype that searches available content looking for specifications that match a user query, and presents the results of the search action to the user. The user may further process the results, for example, by exploring the meaning of definitions or by retrieving extra information. The specification levels of a service are therefore presented to the user as additional information. Our prototype has a web interface that can be accessed via an ordinary web browser.

Our prototype was designed by considering the information, technological and organizational means presented in Table 1. Therefore, we defined the data structures for the prototype in a class diagram (information means), the data flows relating data to IT processes in a dataflow diagram (technological means) and the activities of the process for searching service specifications in an activity diagram (organizational means, especially end users support).

Figure 7 shows how the interactions between the consumer and the service registry of Figure 8 for searching service specifications can be refined.

At the beginning the user is expected to enter and submit a query, but can also choose to stop (arrow to the left from Activity 1.0). If the query is invalid, the system ignores this query and waits for a new one. The number of returned results determines the next action of the system. If there are multiple results, the user is asked to choose from the list of result options. If there is only one result, the user does not have to choose and the system presents the result summary. If there are no results, the system returns to the initial state and waits for a new query. The search query can be adjusted in case the user is not satisfied with the returned list of results, after which the query is again checked for validity. However, when a result is chosen, the system shows simultaneously the result summary, further inspection options and alternatives. The process stops when the user is satisfied, but the user can also choose an inspection option, after which the result for that inspection is shown. This can be repeated while the alternatives and inspection options remain displayed.

Figure 8 shows a dataflow diagram that represents the data stores used to produce the search results. Some activities do not require any specific data in order to be supported, such as validity checks and user input actions, which may be performed without necessarily using data stores.

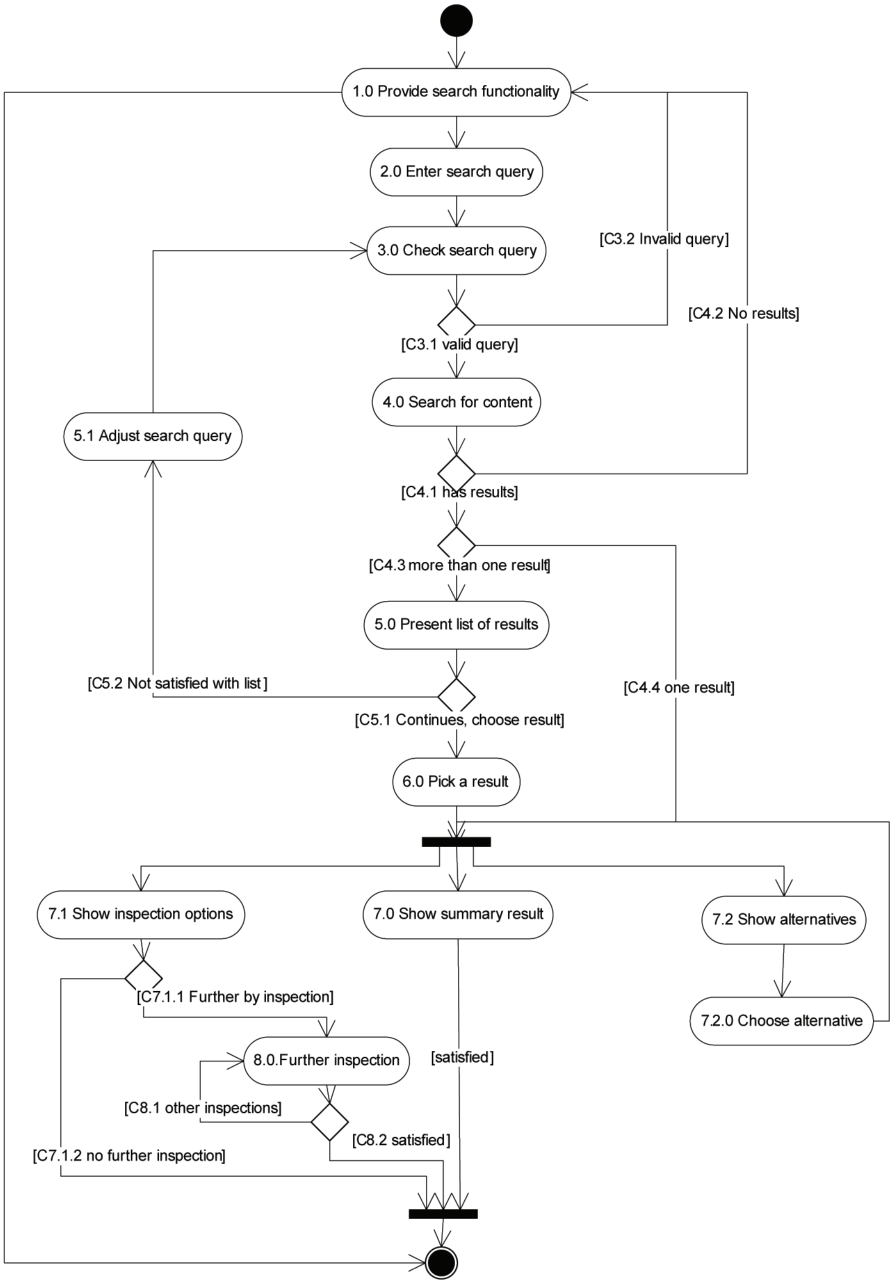
When a request for a search action is retrieved by the web server, the user interface for the search form is created. The search options presented to the user are derived from the appropriate data store. By configuring the data store, the service registry can adjust the options that are displayed. The user submits the search data, and the process uses different data stores to generate the search results. Initially the search data is stored for future analysis. Matching service specifications are derived from the service specifications data store, and ontology mappings are used to determine the relevance of the results and alternatives. When a choice is made from the result list, the summary result for that specification is generated from the service specifications. When the user chooses to inspect the results further (additional information), the relevant specification data is retrieved.

Our prototype implements the process depicted in Figure 7 using the data stores shown in Figure 8. Due to page limitations we omit the class diagram that defines the data structures used in the prototype, which can be found in Van Oostrum (2008).

EXPLOITATION AND EVALUATION

We addressed the ‘Exploitation and evaluation’ layer of Figure 2 by testing our prototype in a case study. In this case study, we exposed our

Figure 7. Activity diagram for the process of searching for service specifications



service registry prototype to experienced IT and business professionals from a consultancy organization, and asked for their opinion on the use of the prototype. We expected that the prototype based on the design as an information service should have a higher user satisfaction than average. An average user satisfaction is defined here as a score of 3 on a five-point scale

Figure 8. Data Flow Diagram for the 'Search design service specification' process

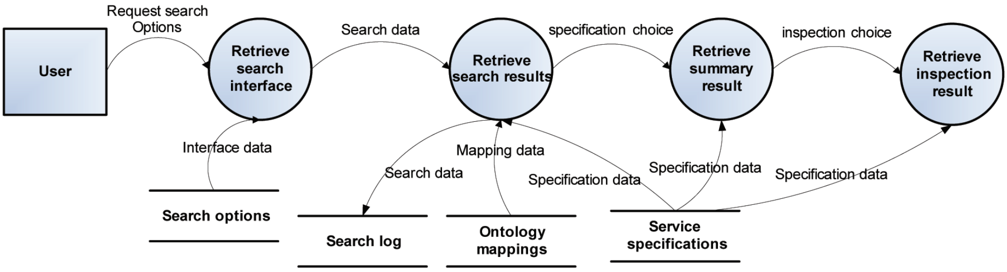
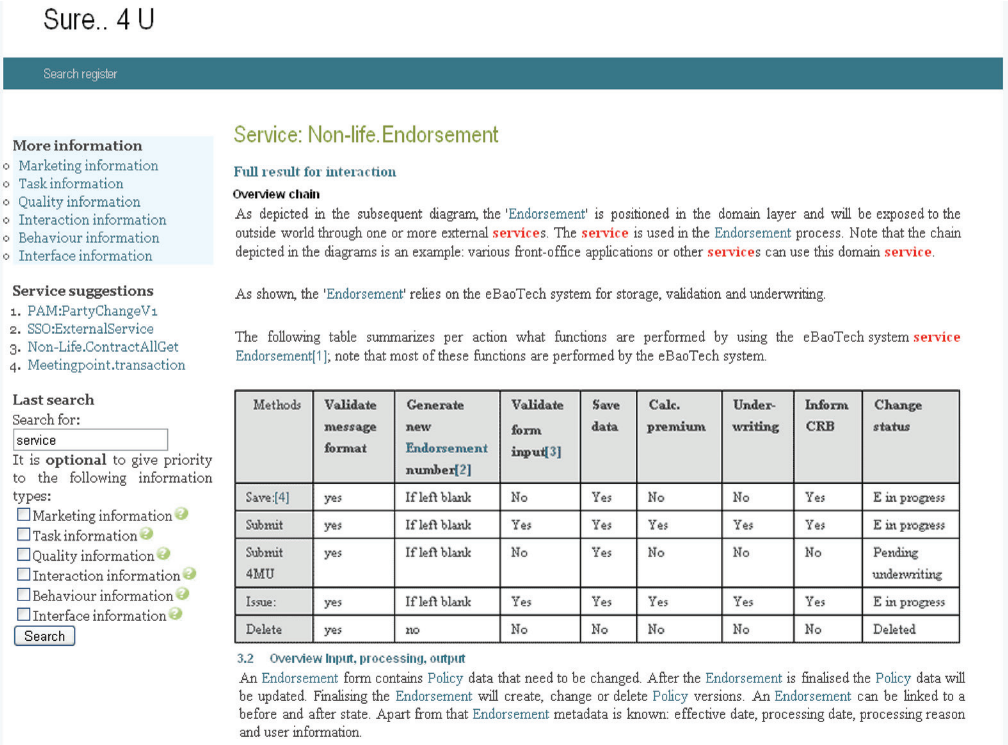


Figure 9. Screenshot of our service registry prototype implementation



Sure.. 4 U

More information
- Marketing information
- Task information
- Quality information
- Interaction information
- Behaviour information
- Interface information

Service suggestions
1. PAM:PartyChangeV1
2. SSO:ExternalService
3. Non-Life.ContractAllGet
4. Meetingpoint.transaction

Last search
Search for:
service
It is **optional** to give priority to the following information types:
□ Marketing information
□ Task information
□ Quality information
□ Interaction information
□ Behaviour information
□ Interface information

Service: Non-life.Endorsement

Full result for interaction

Overview chain

As depicted in the subsequent diagram, the 'Endorsement' is positioned in the domain layer and will be exposed to the outside world through one or more external **services**. The **service** is used in the Endorsement process. Note that the chain depicted in the diagrams is an example: various front-office applications or other **services** can use this domain **service**.

As shown, the 'Endorsement' relies on the eBaoTech system for storage, validation and underwriting.

The following table summarizes per action what functions are performed by using the eBaoTech system **service** Endorsement[1]; note that most of these functions are performed by the eBaoTech system.

Methods	**Validate message format**	**Generate new Endorsement number[2]**	**Validate form input[3]**	**Save data**	**Calc. premium**	**Under-writing**	**Inform CRB**	**Change status**
Save:[4]	yes	If left blank	No	Yes	No	No	Yes	E in progress
Submit	yes	If left blank	Yes	Yes	Yes	Yes	Yes	E in progress
Submit 4MU	yes	If left blank	No	Yes	No	No	No	Pending underwriting
Issue:	yes	If left blank	Yes	Yes	Yes	Yes	Yes	E in progress
Delete	yes	no	No	No	No	No	No	Deleted

3.2 Overview Input, processing, output

An Endorsement form contains Policy data that need to be changed. After the Endorsement is finalised the Policy data will be updated. Finalising the Endorsement will create, change or delete Policy versions. An Endorsement can be linked to a before and after state. Apart from that Endorsement metadata is known: effective date, processing date, processing reason and user information.

for measuring user satisfaction. User satisfaction above average was expected because some use features are implemented in the prototype. High user satisfaction may lead to a better use of the service registry system (Ives, Olson, & Baroudi, 1983), and is a success factor for the service registry. Figure 9 gives a screenshot of our prototype implementation. Some use features are shown on the left bar ('more information' and 'service suggestions'), along with the information on the service that has been found and selected (the Non-life.Endorsement service).

Settings

In the case study we have used descriptions of actual services implemented at the insurance company. This company primarily sells insurances via intermediaries, and has decided to create an SOA architecture to provide services that can be accessed by the intermediaries. The descriptions of these services have been adapted and stored in the prototype.

The prototype could be accessed via the Internet. A group of 89 professionals were invited to participate in this case study. Each participant has first been asked to go to a portal web page and read a short introduction to the assignment before proceeding to the prototype. The participants should perform tasks with the prototype that reflect their actual information needs concerning the search for service information, as they experience in their daily work at the organization. After performing these tasks, each participant has been asked to fill out a short questionnaire to measure end-user satisfaction, using questions derived from the model of Doll and Torkzadeh (1988) and a 5-point rating scale.

The insurance company had no service registry before, but kept the service descriptions in separate documents. We used data from these documents to populate the service registry.

Tasks

Each participant was asked to read an introduction and then to perform two simple tasks that reflect the typical information needs of a service registry end-user, and that yield as a result at least one service specification. In this case study we considered a collection of services of an insurance company. The insurance company uses an authentication application that provides validation, authentication and authorization services via an ExternalService. Intermediaries use these and other services to perform various tasks regarding insurance data. These intermediaries sell insurance contracts (with coverage, obligations and rights, which altogether are called a policy) to consumers. The two simple tasks to be performed by the participant consisted of answering two of the following questions:

- Is there a service to ‘change a consumer characteristic’?
- What is sent as input to the’validation’ method of the ‘ExternalService’ service?
- Is ‘Cancellation of the entire Policy’ one of the supported types of the ‘Endorsement’ service?
- Can the ‘ContractAllGet’ service be used to retrieve ‘Coverage’ details?
- What code is returned by the ‘PartyChangeV1’ service, in case it completes successfully?

ExternalService, Endorsement, ContractAllGet and PartyChangeV1 are some of the services that were registered in the service registry prototype for the purpose of our case study.

After the questions were shown, the participant was instructed to follow a link to the prototype, perform the tasks and return to the screen to fill out an evaluation form. The first part of the form contained the questions derived from Doll and Torkzadeh (1988) to measure the end-user satisfaction. These questions were answered on a five point scale, ranging from 1 = ‘totally not’ to 5 = ‘absolutely yes’. Additionally, the participant was inquired about his/her experience with SOA and specifically with service registries, in order to record the skills or expertise level of the participant. Furthermore, the participant was asked to rate his/her overall satisfaction with the prototype on a five point scale, consisting of 1 = ‘nonexistent’; 2 = ‘poor’; 3 = ‘fair’; 4 = ‘good’ and 5 =’excellent’. Any additional feedback could be given by the participant on a ‘remarks’ text field.

Results

From the 89 invitations, 21 persons have actively participated in the research. From these, 17 participants indicated that they had worked with a service registry before. The majority of the participants rated their familiarity with SOA as high (mean of 3.95 on a 5-point scale), and rated their familiarity with the concept of service registry slightly lower (mean of 3.29 on a 5-point scale).

We are aware that the number of participants is too low to draw significant statistical conclusions. Therefore, we carefully use terms like 'suggests' and 'indicates' when discussing the data. Table 2 summarizes the resulting data.

Table 2 shows the average of the accumulated scores and the average of the overall scores indicated by the participants. Scores average denotes the average score of all 12 questions by the 21 participants (so the average of 252 scores). Overall score denotes the average of the score of the 'overall satisfaction' question. The overall score indicated by the participants is close to the scores average. This suggests that the questions provide a good indication of the user satisfaction. The scores above the 3-point average on a 5-point scale suggest that the participants are more than average satisfied with our prototype.

Table 2 also gives the results for the questions on the factors defined in Doll and Torkzadeh (1988) (contents, accuracy, format, ease of use and timeliness), in terms of the average and standard deviation values. The relatively low standard deviation suggests that the participants rated the content and accuracy generally above 3 points. However, the prototype scores slightly above average for the format and ease of use aspects.

Feedback

Still in the scope of the 'Exploitation and evaluation' layer of Figure 2, we gave the participants an opportunity to give feedback concerning this research. A few participants thought the prototype gave useless details at some service specification levels, while others recognized the usefulness of the division into various information levels (shown in Figure 9 under the 'More information' heading). Some participants would like to know how the prototype would perform in case the search questions were fuzzier.

Although the information in the service specifications was split into the various specification layers, participants still experienced information overload for the full result of a level. This calls for a further reduction of the level of representation of the content, to deliver to the end-user with as few as possible, but yet useful information.

CONCLUSION

The design of a service registry is a complex undertaking due to multiple choices that have to be made about its functionality. By properly scoping the service registry, one can identify the information needs and design requirements of the service registry. We have shown that the design method for information services is useful for developing service registries that fulfill the information needs of its various stakeholders. The design theory forces the designer to approach the service registry by considering its different stakeholders, namely the service providers, management, maintenance personnel and consumers. The design method identifies a design space consisting of layers (design problem, business, process, infrastructure, prototype and exploitation) and aspects (content, use features and revenues) that should be defined for a service registry to fulfill the requirements set by its users. We have also shown how this design theory can be applied, by means of a limited yet representative prototype. Although our case study could give a positive indication of the suitability of our approach, future research could be performed to empirically

Table 2. End user satisfaction scores of service registry users

	Content	Accuracy	Format	Ease of use	Timeliness	Overall score	Scores average
Average	3.52	3.71	3.29	3.36	3.64	3.38	3.50
Standard deviation	0.92	0.59	0.96	0.84	0.81		

determine the appropriateness of the design theory for information services that we applied to the design of more realistic service registries, e.g., by evaluating its effectiveness for service registry development projects in product and process dimensions.

The quality of the service specifications (i.e., the content) stored in the registry is quite important in our approach. Previous attempts to create a public service registry have shown that if the information cannot be trusted for its quality, the service registry is marginally used, or not used at all. Therefore, the organization that considers using a service registry should deploy mechanisms to ensure the high quality of the information and to facilitate its usage by end-users (i.e., use features). The service registry is thus more than just a software system, and also requires aligned organizational structures. A separate organizational unit can be created or some roles have to be assigned to employees in order to run and maintain the service registry. Sponsorship could be arranged in order to deal with costs of running and maintain a service registry.

Our design methodology provides a structured design approach for service registries. Service specifications should be available at different levels, both to enable structured storage of service information and to facilitate the addition of use features. These specification levels cover information needs from service registry consumers, allowing end-users to choose their desired level of specification. Both technical and business users should be able to find their desired information.

This paper assumed that the services are already available, and considered the service registry as a means to facilitate service governance. Equally important is to consider the granularity at which services are defined and how this granularity influences the service registry. This is surely an interesting topic to be investigated in the future.

We built a prototype as a 'proof-of-concept' and performed a case study to assess the suitability of our approach. The case study showed a user satisfaction consistently higher than the 3-point average on a 5-point scale. However, due to the small research population size we are aware that these results should be interpreted with care.

In our prototype, the service registry stores meta-information on the service specifications that we found relevant for the case study, but we have not investigated what kind of information is to be stored and what standards are to be used when specifying a service. Further research should be performed to define a domain ontology for service specifications that consists of service specification concepts and their relationships. This ontology could then be used to search for specifications, to reason on the relevance of results, and to suggest other specifications that use different names for the same concept or that use different but strongly related concepts. Furthermore, if consensus can be reached on the service specification concepts, standards can be produced and products can be developed based on these standards. These should facilitate the interoperability between products and ultimately between organizations (Van der Aalst & Kumar, 2003).

ACKNOWLEDGMENT

The authors are thankful to Ordina System Development and Integration B.V. (the Netherlands) for creating the proper environment for performing this work, and to Vikram Sorathia for his contributions to this paper. The authors also acknowledge the Editor-in-Chief of the journal, Professor John Wang, and the anonymous reviewers for their remarks and suggestions to improve this paper.

REFERENCES

Afshar, M. (2007). *SOA governance: Framework and best practices*. Redwood Shores, CA: Oracle Corporation.

Al-Masri, E., & Mahmoud, Q. H. (2008). Toward quality-driven web service discovery. *IT Professional*, *10*(3), 24–28. doi:10.1109/MITP.2008.59

Alonso, G., Casati, F., Kuno, H., & Vijay, M. (2004). *Web services concepts, architectures and applications*. Berlin: Springer.

Belter, R. (2008). Towards a service management system in virtualized infrastructures. In *Proceedings of the IEEE International Conference on Services Computing 2008* (Vol. 2, pp. 47-51). Washington, DC: IEEE Computer Society.

Bose, R., & Suumaran, V. (2006). Challenges for deploying web services-based e-business systems in SMEs. *International Journal of E-Business Research*, *2*(1), 1–18.

Doll, W. J., & Torkzadeh, G. (1988). The measurement of end-user computing satisfaction. *Management Information Systems Quarterly*, *12*(2), 259–274. doi:10.2307/248851

Fettke, P., & Loos, P. (2003). Specification of business components. In M. Aksit, M. Mezini, & R. Unland (Eds.), *Objects, Components, Architectures, Services, and Applications for a Networked World, International Conference NetObjectDays* (LNCS 2591, pp. 62-75). Berlin: Springer Verlag.

Gorbenko, A., Romanovsky, A., & Kharchenko, V. (2008). How to enhance UDDI with dependability capabilities. In *Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference* (pp. 1023-1028). Washington, DC: IEEE Computer Society.

Gordijn, J., & Akkermans, H. (2001). Designing and evaluating e-business models. *Intelligent E-Business*, *16*(4), 11–17.

Hepworth, M. (2004). A framework for understanding user requirements for an information service: Defining the needs of informal carers. *Journal of the American Society for Information Science and Technology*, *55*(8), 695–708. doi:10.1002/asi.20015

Hevner, A. R., March, S. T., & Park, J. (2004). Design science in information systems research. *Management Information Systems Quarterly*, *28*(1), 75–105.

Ives, B., Olson, M. H., & Baroudi, J. J. (1983). The measurement of user information satisfaction. *Communications of the ACM*, *26*(10), 785–793. doi:10.1145/358413.358430

Lankhorst, M. (2005). *Enterprise architecture at work: Modelling, communication and analysis*. Berlin: Springer.

Larsen, G., & Wilber, J. (2005). *Asset lifecycle management for service-oriented architectures*. Retrieved January 23, 2008, from http://www.ibm.com/developerworks/rational/library/oct05/wilber/

Li, Q., Liu, A., Liu, H., Lin, B., Huang, L., & Gu, N. (2009). Web services provision: Solutions, challenges and opportunities. In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication* (pp. 80-87). New York: ACM.

Luo, J., Montrose, B., Kim, A., Khashnobish, A., & Kang, M. (2006). Adding OWL-S support to the existing UDDI infrastructure. In *Proceedings of the IEEE International Conference on Web Services* (pp. 153-162). Washington, DC: IEEE Computer Society.

Mahajan, R. (2006). SOA and the enterprise: Lessons from the city. In *Proceedings of the IEEE International Conference on Web Services* (pp. 939-944). Washington, DC: IEEE Computer Society.

Papazoglou, M. P. (2008). *Web services: Principles and technologies*. Essex, UK: Pearson Education Limited.

Ran, S. (2003). A model for web services discovery with QoS. *ACM SIGecom Exchanges*, *4*(1), 1–10. doi:10.1145/844357.844360

Samavi, R., Yu, E., & Topaloglou, T. (2009). Strategic reasoning about business models: A conceptual modeling approach. *Information Systems and E-Business Management*, *7*(2), 171–198. doi:10.1007/s10257-008-0079-z

Van der Aalst, W. M. P., & Kumar, A. (2003). XML-based schema definition for support of interorganizational workflow. *Information Systems Research*, *14*(1), 23–46. doi:10.1287/isre.14.1.23.14768

Van Oostrum, A. (2008). *A service register as an information intermediary: towards a structured design approach for service registers*. Unpublished master's thesis, University of Twente, Enschede, the Netherlands.

Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S., & Miller, J. (2005). METEOR-S WSDI: A scalable P2P infrastructure of registries for semantic publication and discovery of web services. *Information Technology Management*, *6*(1), 17–39. doi:10.1007/s10799-004-7773-4

Wijnhoven, F. (2008). Process design theory for digital information services . In Bernard, A., & Tichkiewith, S. (Eds.), *Methods and tools for effective knowledge life-cycle-management* (pp. 533–546). Berlin: Springer. doi:10.1007/978-3-540-78431-9_31

Wijnhoven, F., & Kraaijenbrink, J. (2008). Product-oriented design theory for digital information services: A literature review. *Internet Research*, *18*(1), 93–120. doi:10.1108/10662240810849612

Winkler, M., Cardoso, J., & Scheithauer, G. (2008). Challenges of business service monitoring in the internet of services. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services* (pp. 613-616). New York: ACM.

Womack, R. (2002). Information intermediaries and optimal information distribution. *Library & Information Science Research*, *24*, 129–155. doi:10.1016/S0740-8188(02)00109-3

Yu, Q., Liu, X., Bouguettaya, A., & Medjahed, B. (2008). Deploying and managing web services: Issues, solutions, and directions. *The VLDB Journal*, *17*(3), 537–572. doi:10.1007/s00778-006-0020-3

Luís Ferreira Pires is an Associate Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente, Enschede, the Netherlands since 1994. He obtained his PhD degree in 1994 from the University of Twente and his Master's degree in 1989 from the 'Escola Politécnica da Universidade de São Paulo', São Paulo, Brazil. His research interests include design methodologies for distributed systems, architecture of distributed systems, modeling and specification techniques, middleware platforms and distributed applications. Lately he has been working on the application of model-driven architecture and semantic web technologies to the design of context-aware applications and services. He published around 100 papers in international workshops, conferences and journals. He is currently responsible for the courses 'Programming 1/2', 'Java Middleware Technologies' and 'Service-oriented architecture with Web services' at the University of Twente.

Arjen van Oostrum obtained his master's degree in Business Information Technology (BIT) from the School of Management and Governance of the University of Twente, Enschede, the Netherlands, in 2008. His master's final project has been performed in the scope of an internship at Ordina System Development and Integration B.V., the Netherlands. His main interest is the application of technological solutions to business challenges. He currently works as a ~~trainee~~ for a semi-governmental organization responsible for the storage and processing of information of all Dutch health-care organizations. He also runs a private company that develops advanced web applications.

Fons Wijnhoven is Associate Professor of Knowledge Management & Information Systems at the University of Twente since 2002, and received a PhD in MIS in 1995. He researches the development and exploitation of information services in companies, between companies, and in networks of expertise. Last decade over 50 of his articles appeared in academic journals, and he (co-)authored several books. He teaches in the programs of Business Information Technology, Industrial Engineering & Management, Business Administration, and Public Administration of the University of Twente. He is an associate of the national research school on operations management, BETA, and member of the Center of Telematics and Information Technology of the University of Twente. He has been affiliated to the Helsinki School of Economics (1994), Nijenrode Business School (1998), the Finnish InfWest program (2003-2005), and has been a guest professor in information management at the Wilhelms-University of Muenster (Germany) in 2009.