In: Sebastian Schaffert, Klaus Tochtermann, Tassilo Pellegrini (eds.): Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems, Springer 2009

User-Driven Semantic Wiki-based Business Service Description

Heiko Paoli and Andreas Schmidt and Peter C. Lockemann

Abstract A key factor for success of companies operating in a globalized market environment is a modern SOA-based infrastructure. An essential component of a SOA infrastructure is the central service registry. Current standards for organizing service registries and their implementations are driven by the technical aspects of the infrastructure. When using such technically organized service registries, business users often fail to find the needed information. With the concepts of Web 2.0 in mind, we present a new approach to the organization and implementation of the business registries that are driven by the needs of business users. The paper discusses the problems of the current technically driven approaches, presents an architecture for a business user-driven service registry and introduces an implementation of the architecture using UDDI and Semantic MediaWiki.

1 Introduction

A key factor for the success of companies operating in a global market environment is a flexible communication and information infrastructure that can be quickly and easily adapted to changing needs. Lately, service orientation has evolved as one of the more promising concepts for providing this flexibility [Cearley et al. 2005]. Information infrastructures that follow the paradigm of Service-Oriented Architecture

Heiko Paoli
FZI - Research Center for Information Technologies, Karlsruhe, GERMANY, e-mail: Heiko.Paoli@fzi.de

Andreas Schmidt
FZI - Research Center for Information Technologies, Karlsruhe, GERMANY, e-mail: Andreas.Schmidt@fzi.de

Peter C. Lockemann
FZI - Research Center for Information Technologies, Karlsruhe, GERMANY, e-mail: Peter.Lockemann@fzi.de

(SOA) allow information processes to be defined conveniently and with minimal effort as a succession of calls on available services [He 2003, Huhns and Singh 2005].

Judging from the many trade journals, service orientation does not yet live up to these expectations. We claim as our thesis that the failure is due to service descriptions that are of little help to the business users. Current descriptions have been written by service developers and just cover technical aspects such as service interface, formal parameters, or supported protocols. But this is not the world of the business users who initiate and control the business processes and react to numerous events in them. They need to know which services are available for which business purpose, how these services can be connected, which services have to be replaced when a business process has to be changed or whether new services are needed in order to adapt to new requirements.

As part of the solution we propose differentiating between different stakeholders. The design of information processes should be the responsibility of personnel that understands both, information systems and the business processes (we refer to them as *business analysts*). They need to know what the services have to offer to the business, and they should be able to communicate with the *business users* to map their needs to calls on the services. How these services have been technically implemented should be of little concern to them. The implementation of the services, and their connection to information processes, is the domain of *service developers*.

Service registries should address all stakeholders. Current service descriptions, though, concentrate on the service developers. To include the business aspects in a published service description would be the task of the business analysts. The objective of this paper is to discuss how the analysts can effectively be supported to carry out this task. Any solution should keep in mind that in an environment subject to frequent change, service description cannot be a one-time affair but rather a continuous and collaborative effort among business analysts and service developers [Stojanovic and Dahanayake 2005].

Web 2.0 seems to be an appropriate interaction paradigm in which all stakeholders can be given an active part in service description. This paper presents a new collaborative and lightweight approach to describing services, and shows how business users can take an active part in it, so that a service registry would be able to cover their needs as well.

2 Problem Analysis

As discussed before, service discovery has technical and business ("semantic") facets. The technical part of a service description deals with the syntax of the service interface and is affected by the underlying SOA infrastructure. The semantic part should reflect the business objectives of the service. We examine some of the consequences of the two facets.

2.1 Capturing the Semantics of Business Aspects

The technical part of a service description has always been formulated in a way to make algorithmic processing possible. For the purpose of computer-assisted service discovery the same should hold for the semantic part. Consequently, the business analyst must build a formal model of his or her conceptualization of the business domain, and relate the services to this model.

Take the following example. A business analyst has been given the task to build a new public information portal for flood emergency management. How will he or she find the already published services that might be useful? Suppose the analyst searches for a suitable service under the term of "flood level". Then he or she will in all likelihood miss a service for retrieving the current water level of rivers, even though this would be a good candidate for building the portal. If we had a relation from "flood level" to "water level" and used it in the discovery process, chances would be much higher that more of the appropriate services would be found.

We conclude that traditional information retrieval techniques based on descriptive terms are clearly insufficient and must be augmented by consideration of each term together with its network of somehow related terms.

2.2 Orthogonality of Technical and Business Aspects of the Service Description

Service implementations are technical artefacts and represent technical abstractions from real-world phenomena. Technical descriptions specify how they can, and must be used within a computational environment. Consequently, technical descriptions should only concern the service developers. Likewise, semantic descriptions should solely be of interest to the business analysts and users. Moreover, being an abstraction the same service implementation may be applicable in different business situations and, hence, may have more than one semantic description. Take again the water level service. It may be viewed, and employed, differently by a flood manager, the manager of a river shipping company and the manager of a hydropower plant. And finally, a service may very well have technically been implemented in different ways so that it needs different technical descriptions while the semantic description remains the same.

Consequently, both for technical and application reasons the technical and business aspects of the service description should be kept separate, something that has been known in software engineering as *separation of concerns*[1].

[1] Progr. for Separation of Concerns, `http://www.dmi.unict.it/~tramonta/PSC07/`

2.3 Support of the Dynamic and Collaborative Process of Service Description

Modern business is not a static affair. Consequently, new services may come and all go, while other services must continuously be adapted or applied to new business cases. Continuous change to the business descriptions in the registry is, therefore, a constant challenge for which classical, waterfall-model like approaches that start with business process analysis and end with formal approval, with numerous coordination meetings in between, are ill-suited. In today's interlinked world the flexibility of SOA should be complemented by a more flexible approach where the organization of the business registry should be turned into a collaborative and continuous task along the lines of, say, the Web 2.0 concept.

2.4 Conclusion and Requirements

As we have seen in the problem analysis, a business-oriented service registry should meet three main requirements:

R1 Capture the semantics of business aspects to make services more accessible to business users
R2 Keep technical and business aspects of the service description separate for optimal support of the different user groups
R3 Support the collaborative and dynamic evolution of the service description to accommodate changing needs

3 UDDI as a foundation

UDDI is practically the only standard for advertising services by service registries. The ambitious goal of UDDI was to establish a world-wide service registry to create a world-wide market of services and enable small and unknown companies anywhere in the world to offer their innovative services to customers on the other side of the globe. Therefore we should try to stay with UDDI as the basis of our registry unless UDDI completely fails to accommodate the requirements R1 through R3.

Figure 1 gives a condensed overview of UDDI. Central to UDDI is the UDDI registry. The registry points to the service description (WSDL) and the service itself. The description of a published service provided by WSDL should enable the service consumer to use the service via the underlying technical infrastructure. This description is therefore related to the technical interface of the service, describing syntactically its operations, formal parameters, message types, and supported protocols.

UDDI indeed provides a mechanism for augmenting the service description by metadata, although the mechanism seems fairly cumbersome for business analysts and users. The metadata take the form of name/value pairs that are stored in the technical Model (tModel) of UDDI. The name part of the tModel represents the namespace of any data structure which is to be used to characterize the service, whereas the value part is a unique pointer to the referenced data structure. The idea behind this approach is to categorize the registered services by standardized and uniformly known global category systems, such as the North American Industry Classification System (NAICS). The category system has been criticized as insufficient and the UDDI data model as very limited [Nickull et al. 2006]. Indeed it seems far from satisfying requirement R1 because the category system is much too coarse to describe the services, and also too large for a user to become familiar with and to select the terms appropriate for a given situation. In addition the effort in manpower and time is inordinate to continuously develop huge and global category systems [Zimmermann et al. 2004]. To conclude, the current UDDI concept seems little supportive of business semantics.

The concept of tModel seems to go some way towards requirement R2, though. And indeed, UDDI supports several user roles. The top left-hand corner of Figure 1 shows three user groups of the service registry: developers, business analysts and administrators. Administrators deal mainly with the technical management of the registry and the published services, and provide technical support for the other user groups but do not create or employ new services themselves. However, the two other groups, developers and business analysts, match two of our own stakeholders. But as far as service description is concerned, all user groups are treated alike: There is just one common description method.

Requirement R3 is not addressed by the concept of UDDI at all. UDDI does not care whether the global category systems remain the same or not. Therefore, the process of standardization of category systems is outside the scope of UDDI. Similarly, if a service or its application is changed, it is left to the publisher whether and how to adjust the service description. UDDI does not foresee any explicit support.

Fig. 1 Implementation and usage of a SOA with UDDI (Source [Shen 2004])

To summarize, UDDI as a concept seems well organized to support requirements R1 and R2. It offers no direct support for R3, but nor does it place obstacles in the way. On closer examination, though, even the support of R1 and R2 with the tModel as the only mechanism seems rather poor.

4 Related Work

4.1 Business Semantics (R1)

In [Sivashanmugam et al. 2003] WSDL is the industry standard for describing Web services while UDDI is the industry standard for advertising them. The authors pursue the general objective of automated discovery of Web services taking semantics into account. They propose that a DAML+OIL ontology be used to annotate WSDL message parts in order to add the necessary semantics to the Web service description. WSDL itself is extended by new markup tags which allow to attach the semantic description in the form of the preconditions and effects of a Web service. By the time a service is published to UDDI, the extended WSDL description is mapped to the tModel where it becomes accessible to the discovery process. Sivashanmugam et al. also develop a three-phase algorithm for the discovery process. At the start of a process a template is generated into which the service requirements are entered. In the first phase services are matched by functionality (service operations) and then the result is ranked in the following phases on the basis of semantic similarity of input/output parameters and preconditions/effects.

The work seems to go a long way towards R1. We note, though, that requirement R2 is poorly met: The semantics are entirely embedded in WSDL and thus cannot be separated from the technical description. Further, the semantics are expressed in notations unnatural to the business user.

A bit earlier, Paolucci et al. took a similar approach [Paolucci et al. 2002]. They present in greater detail an algorithm for matching service requests to advertised services based on semantic descriptions in the form of ontologies. From the point of view of R2 their approach seems somewhat more advanced, since the DAML-S semantics are kept on a semantic layer. By the time a service is advertised, a DAML-S/UDDI translator constructs a standard UDDI description and stores it in the UDDI data model while the semantics for the matching algorithm is sent to a DAML-S matching engine and stored there with a reference to the constructed UDDI description. In the discovery process the stored semantics in the DAML-S matching engine is used while the DAML-S/UDDI translator provides the dependent UDDI descriptions. Unfortunately, no application experience is discussed, but it seems doubtful that business users would feel comfortable with the semantic description or would consider the approach transparent enough to evaluate the outcome of their search.

Even earlier, McIlraith et al. already employed the DAML family for semantic markup of Web Services [McIlraith et al. 2001]. Their objective was different, though: They wished to automate the discovery, execution, composition, and interoperation of Web services for the use in multiple agent systems. With the help of ConGolog - a programming language for robot systems - it should be possible to write generic procedures, e.g., a generic procedure to plan a business travel, without knowing which services are currently available and how they should be invoked to execute the procedures. If an agent wants to use a generic procedure, appropriate services are discovered, composed and executed automatically. Again, the discovery process is supported by ontologies. Since everything is automated the use of a common method for technical and semantic aspects is a requisite rather than an obstacle.

4.2 Separation of Aspects (R2)

The discussion in the previous section shows that separation of concerns is on everyone's mind but seems poorly executed from a business application point of view. Separation of concerns is a widely held philosophy in software engineering, but there the technical aspects predominate, and the experts involved are technical people. Still one may learn from the general model by Bergmans et al. for composing systems from multiple concerns [Bergmans et al. 2001]. The authors introduce a number of requirements for design-level composability, and define a category of composability problems that are inherent for given composition models. One result are criteria when separation of concerns should be applied to reduce the complexity of software by composing independent components, and when it should be avoided because of composition anomalies. We conclude that our approach does not fall into the category of composition anomalies so that requirement R2 is indeed justified.

4.3 Collaborative Service Description (R3)

Collaborative work in general, and collaborative authoring in specific, is nothing new. However, since we wish to make use of standards – such as UDDI – we need to employ standards for the collaboration as well. Such a standard is MediaWiki where categories can be assigned to articles in order to support searching and navigating through its content. Krötzsch et al. extend this concept to links between articles so that they become machine-processable [Krötzsch et al. 2006]. Links between articles can be viewed as named relations, and articles can have named attributes. Both can be used for navigation and searching by an embedded query language. The language can also be used to create dynamic articles, e.g., to have an article in which all services related to "water level" are listed. This article is automatically updated when a new article about such a service is created or when an already published article is deleted.

[Krötzsch et al. 2006] seems to confirm that a Semantic MediaWiki is ideal as a frontend for business analysts because it is easy to use, allows adaptation to dynamic changes in a collaborative way and, moreover, is a suitable framework for the semantic needs of R1. Besides adjusting service descriptions, one mainly collaborative task is the continued development of the ontologies. [Zacharias and Braun 2007] reports on the engineering of lightweight ontologies by using tagging mechanisms. The idea behind this approach is that interesting information is shared within a community, which is then tagged by the latter to categorize it. Concepts of a lightweight ontology can then be derived from the used tags. The ontology is constructed and changed in a collaborative and Web 2.0-like way.

5 A Comprehensive Approach to Business Service Description

5.1 Basic architecture and workflow

We start with requirement R3. Similar to the suggestion in Section 4.3 we make use of Web 2.0. More specifically, we take a Semantic MediaWiki-based approach to the collaborative development of the business registry. To meet requirement R2 we decide to stay with UDDI for the technical registry and to add the Wiki solution as a front end to UDDI. Finally, to satisfy requirement R1 we follow the approach of Section 4.1 and employ ontologies to capture the network of related terms. In particular, our aim is a lightweight ontology that can be easily handled by business experts without extensive training in ontology engineering. In contrast to the approach of Section 4.1 we do not extend the UDDI data model but rather use the light-weight ontology with the Semantic MediaWiki. Contents of the UDDI Registry are dynamically rendered by an extension of the Semantic MediaWiki. We refer to our solution as an *Extended Semantic MediaWiki*.

Figure 2 shows the system architecture. It consists of four main components: a UDDI-based technical registry, a Semantic MediaWiki-based business registry, an ontology server and an ontology engineering component. The figure also indicates the basic workflow within the architecture. A software developer as a service publisher can use any UDDI-compatible client to publish a new service into the registry, which may also include a technical description like a WSDL file in the case of a Web service. In addition to the technical description, the software developer may add some keywords based on the ontology in order to roughly categorize the business use of the Web service. The content of the UDDI Registry is dynamically embedded into the content of the Semantic MediaWiki, which forms the business-oriented registry. The keywords chosen by the software developer are used as an initial categorization for the service. From now on business users can search or navigate along the contents of the Semantic MediaWiki, add additional information to the dynamically generated pages, or create new pages. A Semantic MediaWiki is chosen to make the contents of the business registry machine understandable and to add implicit facts with the help of an ontology server. The ontology engineering component allows the business users to adapt the used business ontology to their needs in a lightweight and collaborative way.

We will discuss in a bit more detail the steps that have been numbered in Figure 2.

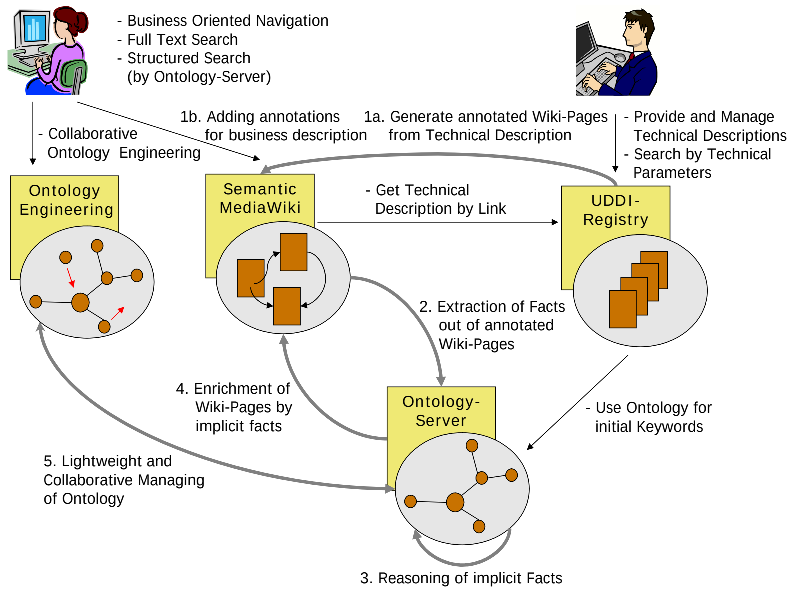

Fig. 2 System Architecture: Combining UDDI with a Semantic MediaWiki

5.2 Ontology

We observe from Figure 2 the central role of the ontology. Hence, we give a very brief outline. The left-hand side of Figure 3 gives an example of the organization of our ontology. The top level part provides the domain-independent concepts such as the terms *Concept*, *Business Object* and *Service*. These are refined to a network of concepts of the business domain of which Figure 3 just shows three examples, the terms *water*, *water level* and *water gage information*.

In a collaborative environment the presentation of ontologies for effective and efficient use by the business analysts is particularly important. Presentation of the business registry is in the form of Wiki pages, with relations between concepts mapped to semantic links supported by the Semantic MediaWiki (right-hand side of Figure 3). In the example, *water* is a top level (business) concept while *water gage* is a business object concept, and *water gage information* is of type *service* and stands for a published and reusable service in the SOA infrastructure which will return a *water gage*.

We use OWL-Lite as the ontology description language. Currently we use KAON2 as a reasoner, but any other compatible reasoner should also be possible [Motik and Sattler 2006]. To make the ontology both persistent and generally available it is stored in a relational database from where it can be retrieved by the Semantic MediaWiki, the ontology server, the ontology engineering component, and the UDDI registry.

5.3 Annotation of Wiki pages

A service such as *water gage information* is initially entered into the system by its developer. He or she publishes it to the UDDI registry together with a technical WSDL description, and is encouraged to augment it by intuitive keywords found in the ontology. Together with the publication a Wiki page is generated for the service, and automatically annotated with the aforementioned keywords as well as semantic links that are obtained from the relations of the general UDDI data model. The "Business Entity" element of the UDDI data model denotes the business analyst who is responsible for the business description so that the analyst may now be notified of the new service.

Business analysts can create new Wiki pages or modify existing ones (including generated Wiki pages) for the purpose of adding further annotations. The annotation of Wiki pages can be carried out by means of such Semantic MediaWiki features as semantic links, semantic attributes, and inline queries (to embed dynamic content). Many annotations can be obtained from the ontology by navigating through it and extracting further facts, or by using the reasoner to derive implicit facts or some of the semantic links. For example, on the left-hand side of Figure 3 the solid arrows represent relations that are explicitly available from the ontology (*hasType*, *belongsTo*, *provides*), while the dashed arrows represent relations that are implicitly available because of reasoning through the ontology server.

Not only does our approach satisfy requirements R2 and R3, but it clearly does so with great benefit to the two stakeholders of business analyst and service developer. A business analyst can concentrate on the business description and freely organize and annotate the Wiki pages. For example he or she may express the business context of a service, e.g., business use cases, business value etc. The business description is limited neither by the (technical) data model of UDDI nor the facilities of WSDL (that would allow us to describe a service only along its technical interface, e.g., operations, input, output parameters). On the other hand the UDDI registry remains compatible to current SOA implementations and allows developers to use their favorite UDDI tool to publish the technical description of newly implemented services.

5.4 Service discovery

Other than in Section 4.1 we do not foresee automatic service discovery. Rather both the business analyst and the business user discover appropriate services by navigating through the ontology. This explains the emphasis we give to the presentation via

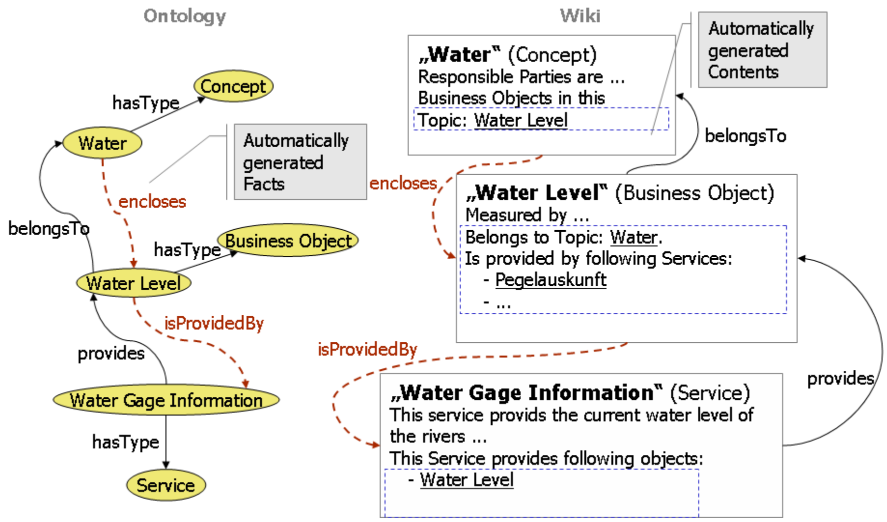

Fig. 3 Organization and presentation of the business registry together with dynamically embedded UDDI entries and implicit facts

Wiki pages. Take again the right-hand side of Figure 3. Note that much of the page contents for all terms is automatically generated. In particular, business object pages list all relevant services. For example, for an overview page on *water level* all *water gage* services are listed. If a new service is published which also returns a *water gage*, it will be automatically listed on the *water level* page without any additional manual intervention.

Consequently, our approach satisfies requirement R1 as well. The proposed organization of the business registry and the use of an ontology which is well known to the business analysts provides a familiar and easy-to-use environment for them. The business registry supports navigation along business objects for discovering needed services. The use of an ontology server together with the domain ontology enables a business-oriented search, e.g., a search for all services which provide a *water level*. The use of dynamic Wiki-Pages makes it possible for business analysts to build individual and well adapted points of informations.

5.5 Lightweight Ontology Engineering

In the dynamic business environment that we postulated in Section 2.3 the ontology itself is bound to frequently change as well. Rather than entrusting a central authority with modifying the ontology we rely on the combined and distributed competency of all business analysts, and perhaps even users. Accordingly, we let the ontology evolve in collaboration of the business experts whenever one sees the need. Since we cannot expect the analysts to be experts in building ontologies, the engineering of the ontology should be made as simple as possible.

We ease the task in two ways. For one the ontology is visualized as a graph, and all modifications can be easily done by dragging and dropping the nodes of the visual presentation rather than in some formal language. Second, the range of possible modifications is restricted (hence the name “lightweight engineering”). It is possible to create alternative labels for a concept and choose a preferred label for it. Concepts can only be connected via *broader-narrower* and *related* relations. By using Wiki pages all modifications to the ontology are immediately seen by all other business analysts.

6 Implementation

Figure 4 shows our implementation of the service registry. It consists of a central relational database, which holds the UDDI entries, the Semantic MediaWiki pages and the ontology. On top of the relational database we have a J2EE application server and an HTTP server with PHP support. The J2EE application server represents the technical UDDI-compatible registry, which is realized through three components: a UDDI framework to support the UDDI API (which enables technical descriptions), a SOAP Engine to support the UDDI protocol, and a UDDI browser to view the contents of the UDDI registry and publish new services. Our implementation is fully compatible to standard UDDI that explicitly allows publishers to use their own UDDI browser if they wish to (shown as the UDDI browser component at the bottom of Figure 4). The HTTP Server with PHP support represents the business-oriented registry realized through an extended Semantic MediaWiki component - the extension is necessary to support the automatic generation of content from the UDDI registry. For ontology engineering we use the existing tool SOBOLEO, a Web-based implementation of a Simple Knowledge Organisation System [Zacharias and Braun 2007].

7 Experiences and Conclusions

The work presented in this paper has its origin in a project that was financed by the Ministry of Environment of Baden-Wuerttemberg. The environmental administration of Baden-Wuerttemberg has a long experience with environmental information systems in service oriented architectures. At the moment a redesign to a modern SOA-based infrastructure is planned by the State Institute for Environment, Measurements and Nature Conservation on behalf of the Ministry of Environment. The main objective is to provide all relevant parts of the system as services by a registry, and it should be possible to add a wide though unknown range of the services in the future. The system should be capable of handling hundreds of business users and ser-

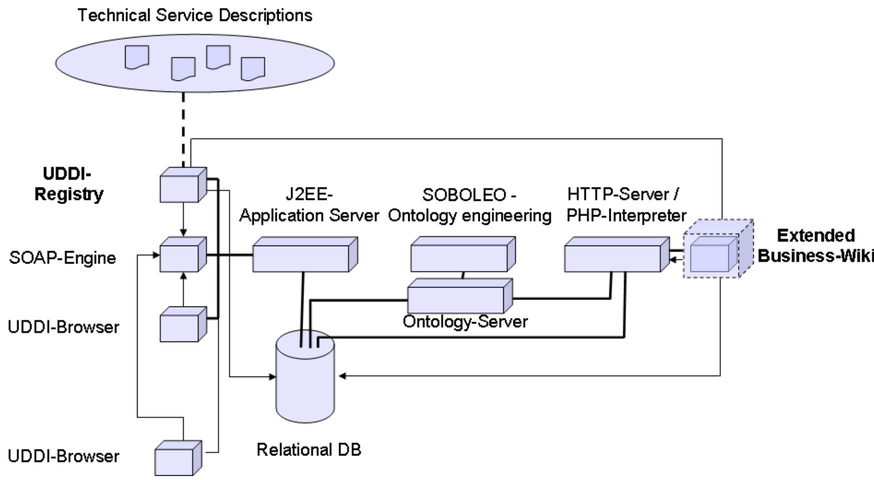

Fig. 4 Implementation of the service registry

vice developers. To avoid duplicate work and to make all published services transparent to all business users a business oriented service registry seemed essential. The initial ontology we have used is based on an already existing and widely used taxonomy developed for the environmental information system of Baden-Wuerttemberg. The technical infrastructure as described above was developed in close communication with more than 10 representatives of business analysts and 5 representatives of developers, and was rolled out for a first testing period in April of 2007. First feedback by users sounds encouraging.

The thesis underlying our work is that service orientation will become widespread only if services can be discovered and employed with ease not just by service developers but also by business analysts. We have translated the needs to three requirements, the separation of technical and semantic descriptions, natural use of the semantic descriptions by business people, and a collaborative approach to dealing with the business dynamics. First experiences seem to support our thesis for the narrow scope of environmental information systems. What is definitely needed is more systematic and wider ranging empirical studies before we can be sure that our approach is an important step in overcoming the still existing doubts on the effectiveness of service-oriented architectures.

References

[Bergmans et al. 2001] Bergmans, L., Tekinerdogan, B., Glandrup, M., Aksit, M.: "Composing Software from Multiple Concerns: Composability and Composition Anomalies"; in Proceedings of International Conference on Software Engineering 2001, Toronto, Ontario, Canada, 2001, http://amidst.ctit.utwente.nl/publications/icse2001_

bergmans.pdf
[Bloomberg 2005] Bloomberg., J.: "The Four Pillars of Service-Oriented Development"; zapthink, 2005. http://www.zapthink.com/report.html?id=ZAPFLASH-2005418 as seen on 2006-09-20
[Brickley and Miles 2005] Brickley, D., Miles, A.: "SKOS Core Vocabulary Specification"; Techreport, W3C Web Site, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102 as seen on 2005-11-30
[Cearley et al. 2005] Cearley, D., Fenn, J., Plummer, D.: "Gartner's Positions on the Five Hottest IT Topics and Trends in 2005"; Gartner Web Site, 2005. http://www.gartner.com/DisplayDocument?doc_cd=125868 as seen on 2006-09-20
[Czarnecki 1996] Czarnecki, K.: "'Separation of Concerns' - objektorientierte Frameworks und das generative Paradigma"; In OBJEKTspektrum Nr. 6, Nov./Dez. 1996. http://www-ia.tu-ilmenau.de/~czarn/concerns.htm
[Decker et al. 1998] Decker, S., Brickley, D., Saarela, J., Angele, J.: "A Query Service for RDF."; in Proceedings of QL, 1998.
[Fayyad et al. 2001] Fayyad, U., Grinstein, G., Wierse, A.: "Information Visualization in Data Mining and Knowledge Discovery"; Morgan Kaufmann, 2001.
[Fensel et al. 1998] Fensel, D., Decker, S., Erdmann, M., Studer, R.: "Ontobroker: The very high idea"; in Proceedings of the 11th International Flairs Conference (FLAIRS-98), Sanibal Island, Florida, 1998.
[Garlan 2000] Garlan, D.: "Software Architecture: a Roadmap"; in: ICSE - Future of SE Track, 2000. http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalgarlan.pdf
[He 2003] He, H.: "What Is Service-Oriented Architecture"; XML.COM, O'Reilly, 2003. http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html as seen on 2006-09-20
[Hepp 2006] Hepp, M.: "Possible Ontologies: How Reality Constraints Building Relevant Ontologies"; in: IEEE Internet Computing, 2006.
[Huhns and Singh 2005] Huhns, M., Singh, M.: "Service-Oriented Computing: Key Concepts and Principles"; in: IEEE Internet Computing, vol. 09, no. 1, pp. 75-81, 2005.
[Jones and Morris 2005] Jones, S., Morris, M.: "A METHODOLOGY FOR SERVICE ARCHITECTURES"; OASIS Web Site, 2005. http://www.oasis-open.org/committees/download.php/15071/A\%20methodology\%20for\%20Service\%20Architectures\%201\%202\%204\%20-\%20OASIS\%20Contribution.pdf as seen on 2006-09-20
[Kifer et al. 2005] Kifer, M., Bruijn, J., Boley, H., Fensel, D.: "A Realistic Architecture for the Semantic Web."; in Proceedings of RuleML, pp. 17-29, 2005.
[Kiryakov et al. 2003] Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: "Semantic Annotation, Indexing, and Retrieval."; in Proceedings of International Semantic Web Conference, pp. 484-499, 2003.
[Krafzig et al. 2004] Krafzig, D., Banke, K., Slama, D.: "Enterprise SOA. Service-Oriented Architecture Best Practices"; Prentice Hall, 2004.
[Krötzsch et al. 2006] Krötzsch, M., Vrandecic, D., Völkel, M.: "Semantic MediaWiki"; In Proceedings of International Semantic Web Conference, Lecture Notes in Computer Science, vol. 4273, pp. 935-942, Springer, 2006. http://dblp.uni-trier.de/db/conf/semweb/iswc2006.html\#KrotzschVV06
[McIlraith et al. 2001] McIlraith, S., Son, T., Zeng, H.: "Semantic Web services"; In IEEE Intelligent Systems (Special Issue on the Semantic Web), March/April 2001.
[Motik and Sattler 2006] Motik, B., Sattler, U.:"A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes"; In Proceedings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2006), Phnom Penh, Cambodia, November, 2006
[Nickull et al. 2006] Nickull, D., McCabe, F., MacKenzi, M.: "SOA Reference Model TC"; OASIS Web Site, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, as seen on 2006-09-20.

[Paolucci et al. 2002] Paolucci, M., Kawmura, T., Payne, T., Sycara, K.: "Semantic Matching of Web Services Capabilities"; In First Int. Semantic Web Conf., 2002.

[Schaffert 2006] Schaffert, S.: "IkeWiki: A Semantic Wiki for Collaborative Knowledge Management"; In Proceedings of WETICE, pp. 388-396, IEEE Computer Society, 2006. http://dblp.uni-trier.de/db/conf/wetice/wetice2006.html#Schaffert06

[Shen 2004] Shen, Z.: "UDDI v3.0 (Universal Description, Discovery and Integration)"; OASIS Web Site, 2004. http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.pdf

[Sivashanmugam et al. 2003] Sivashanmugam, K., Verma, K., Sheth, A., Miller, J.: "Adding semantics to web services standards"; In Proceedings of the 1st International Conference on Web Services (ICWS'03), 2003.

[Stojanovic and Dahanayake 2005] Stojanovic, Z., Dahanayake, A.: "Service-Oriented Software System Engineering, Challenges and Practices"; Idea Group Publishing, 2005.

[Sure et al. 2003] Sure, Y., Angele, J., Staab, S.: "OntoEdit Multifaceted inferencing for ontology engineering"; in: Journal on Data Semantics, vol. LNCS 2800, pp. 128-152, 2003.

[Zacharias and Braun 2007] Zacharias, V., Braun, S.: "SOBOLEO - Social Bookmarking and Lighweight Engineering of Ontologies"; in Proceedings of 16th International World Wide Web Conference (WWW 2007), Workshop on Social and Collaborative Construction of Structured Knowledge (CKC), Banff, Alberata, Canada, 2007. To appear. http://www.fzi.de/KCMS/kcms_file.php?action=link&id=678

[Zimmermann et al. 2004] Zimmermann, O., Krogdahl, P., Gee, C.: "Elements of Service-Oriented Analysis and Design: An interdisciplinary approach for SOA project"; IBM DeveloperWorks, 2004. http://www-128.ibm.com/developerworks/library/ws-soad1/