Economic Reflections on Managing Web Services Using Semantics

Frank Wolff[1], Daniel Oberle[2], Steffen Lamparter[2] and Steffen Staab[3]

[1] ICB Information Systems and Enterprise Modelling
University of Duisburg-Essen - Germany
frank.wolff@uni-duisburg-essen.de
[2] Institute AIFB
University of Karlsruhe - Germany
lastname@aifb.uni-karlsruhe.de
[3] ISWeb
University of Koblenz-Landau - Germany
staab@uni-koblenz.de

Abstract: In Web Services Management different approaches are discussed. In this paper we analyse three options for Web Service Management: a) Web Services standards descriptions and tools (WS*), b) Semantic-supported and c) Automatic semantic Web Services Management. For the analysis of the differences we identify a number of use cases and investigate the information required for the management of Web Services. The three discussed options result in varying processes with considerable differences in the work efforts. As they involve different levels of investment which only repay on a number of similar WS-applications the discrimination between them is not straightforward. A number of factors must be included in the analysis. Based on the cost arguments and the actual status of standards we consider Semantic-supported management of Web Services to be the most promising option for near future.

1 Introduction

Different Web Service standards, WS*, factorize Web Service management tasks into different aspects, such as input/output, workflow, or security.[1] The advantages of WS* are multiple and have already benefited some industrial cases. WS* descriptions are exchangeable and developers may use different implementations for the same Web Service description. However, the disadvantages of WS* are also visible yet: Even though the different standards are complementary, they must overlap and one may

[1] The standards have been proposed by different organisations like the W3C, OASIS, WS-I, BPMI and other groups from industry. Apart from the basic technical standards a number of standards have been devised to support the development, deployment and management of Web Services. Some of the main standards for this purpose are WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration), WS-Policy Framework, WSS (Web Services Security), BPMN (Business Process Modelling Notation) and WSBPEL (Web Services Business Process Execution Language).
See: http://www.oasis-open.org, http://www.w3C.org, http://www.ws-i.org, http://www.bpmi.org .

produce models composed of different WS* descriptions, which are inconsistent, but do not easily reveal their inconsistencies. The reason is that there is no coherent formal model of WS* and, thus, it is impossible to ask for conclusions that come from integrating several WS* descriptions. Hence, discovering such Web Service management problems or asking for other kinds of conclusions that derive from the integration of WS* descriptions remains a purely manual task of the software developers accompanied by little to no formal machinery. Thus, at the economic/technical level WS* descriptions seem to reduce costs for managing software, but the question is whether a more innovative approach couldn't reduce management costs even further. In the following we refer to this approach as '*WS* manual Management'*.

Researchers investigating "Semantic Web Services" have clearly articulated these shortcomings of WS* standardizations and have been presenting interesting proposals to counter some of them [MSZ01]. The core of their proposals lies in creating semantic standards. Their principal objective is a wide-reaching formalization that allows full automation of the Web Service management tasks such as discovery and composition. Therefore, we will call it '*Automatic semantic Management'*. Again, the potential advantages are obvious; but there are also disadvantages: First, at the purely technical level the machinery for the right kind of powerful reasoning still needs to be clarified. Second, at the economic/technical level fully automatic reasoning needs rather fine-grained modelling which comes at a certain cost, – even if the purely technical problems may be solved.

As a third option, we put forward '*Semantic-supported Management'*. This does not tackle full automation of all Web Service management tasks. In fact, we claim that the full breadth of Web Service management requires an understanding of the world that is too deep to be modelled explicitly. Instead, we see the need to do better than with current WS* approaches, but we see a more passive role for reasoning than argued for in many approaches towards 'Automatic semantic Management' for Web Services. In our paradigm, reasoning support is added where requested by the needs of the developers who must cope with the complexity of Web Service integration and WS* descriptions. [JoRa05] They could use valuable tools for integrating previously separated aspects, but they rely on their own abstracting understanding (at which they arrive with the help of the semantic tools) of the situation rather than at full automation of Web Services management.

The separation of the three options is not completely clear cut but it reflects some main division in the discussion on the management of Web Services. So we distinguish three alternative options for Web Services Management. They have considerable consequences on the costs and the attainable benefits.

1. *WS* manual Management* is mostly technically focussed and deals with the different issues of Web Service management separately using the appropriate standards (option 1).

2. *Automatic semantic Management* builds on a clear representation of the relevant technical and domain facts. They are stored in an ontology which on demand decides which service is responsible for a task inside an application process (option 2).

3. *Semantic-supported Management* is based on an integrated semantic model (usually an ontology) for the application domain and thereby covers the technical and the business side. But it is pragmatic in so far, as it does not define everything in detail. It helps to establish a mixture of directly deducible actions and the provision of facilitating information for human developers and administrators (option 3).

It is the purpose of this paper to clarify what kind of objectives could and should be targeted by semantic modelling of Web Services. Therefore we break up the overall task of constructing applications based on a number of Web Services and the task of supporting their actual usage. This also helps to identify important characteristics of ontologies necessary to assist effectively relevant work activities.

Like every kind of modelling, semantic modelling will also cause some costs. To justify these expenses, they must be compensated by higher benefits [WF05]. These benefits depend on a number of different issues. Their analysis requires a closer investigation of the requirements of the Web Services, on one hand, and on the environmental characteristics of the processes in WS-Development and Management, on the other hand. In the coming years the importance of Web Service-based applications will increase substantially. [TG05] This development will be driven by the increased division and specialization of services and by the growing trend to integrate supply chains.

In this paper we look for decision criteria to be evaluated in a specific case which of the three different WS-management approaches is economically most promising. Anyway, an easy direct beforehand calculation of costs and benefits is not possible. [EDBS05] The underlying processes of Web Services management are too complex and interdependent to be open to an easy and direct estimation.

Considering the practical use of Web Services and Semantic technologies one will note a wide gap between the statements stressing their importance and their absence in daily practice. The following reasoning on the underlying technical and economical factors will show that it may take some time until ontologies and familiar semantic technologies will be able to deliver all their promises. Nevertheless, a large part of industry has invested in Web Services. As the basic needs and claims for Web Services will remain valid and these Investments will not been thrown away easily we assume that there will be a lasting commitment. But it will take much longer until Web Services really become mainstream practice. So we consider a long-term perspective appropriate emphasising an evolutionary use of ontologies and semantic modelling for Web Services.

2 Information Structure in Semantic Models for Web Services

In order to realize semantic management of Web Services we have to model service profiles, service taxonomies, policies, workflow information, interface descriptions as well as quality of service information.

Available semantic modelling technologies provide powerful means that support us here: Ontologies. They formalize concepts and concept relationships (associations) very similar to conceptual database schemata or UML class diagrams. However, ontologies typically feature logic-based representation languages that come with executable logic calculi. The calculi allow us to reason and query at runtime [SS04].

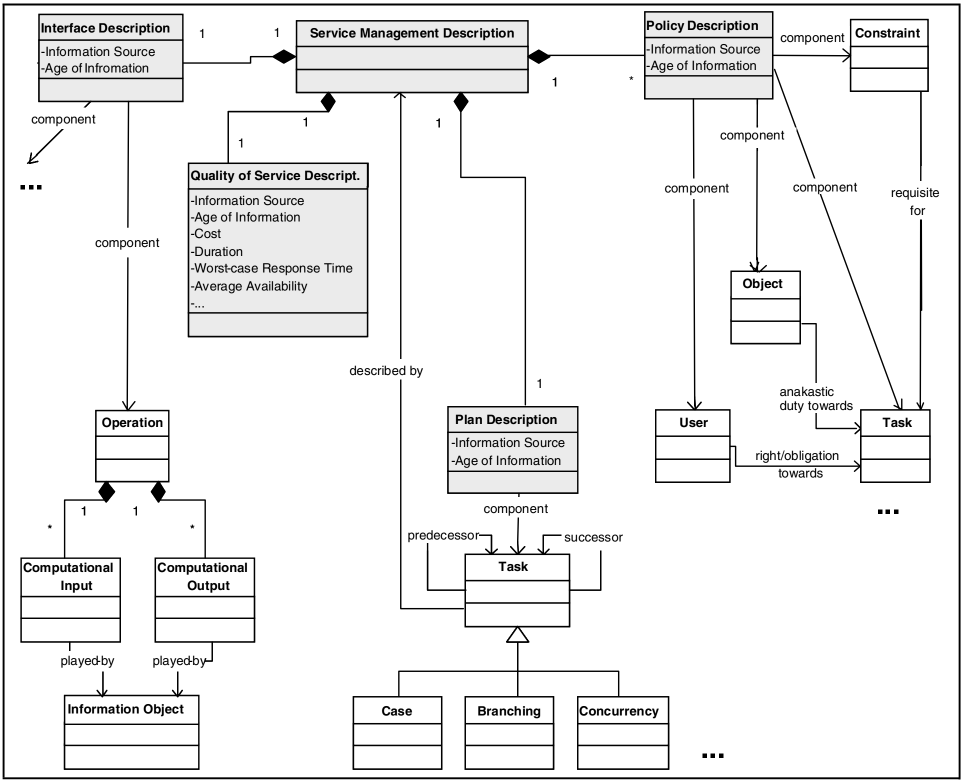

Figure 1: Sketch of the Core Ontology of Services[2]

For semantic management of Web Services we need to formalize an ontology that meets the aforementioned modelling requirements, thus weaving together aspects from multiple WS* descriptions.[OLES05] Querying and reasoning will enable us to ask for conclusions that come from integrating several WS* descriptions — e.g. predicting or observing how Web Services interact, (might) get into conflict, (might) behave, etc. — what had to be done manually before.

Figure 1 presents a sketch of a Core Ontology of Services that models all the necessary aspects. Some parts are already captured by the DOLCE foundational ontology [OGG02], the Ontology of Plans [GBC04] and the Core Legal Ontology [GST04]. The presented ontology of services reuses them and adds some other parts (for details see [OLGV05]). In the following section separate use cases are identified which predominantly concentrate on one aspect of the ontology.

---

[2] A prototype of the ontology is maintained at http://cos.ontoware.org .

3 Use Cases for Semantic-supported Management of Web-Services

There are a number of promising use cases for exploiting semantic modelling. The investigation reveals benefits that can be derived from what kind of semantic modelling of Web Services, when and for what purposes. In particular, each use case will answer the following questions:

Question 1 Who uses the semantic descriptions of Web Services?

We see two major groups of users constituted by (i) software developers and (ii) administrators. These two groups of users have the need to predict or observe how Web Services interact, (might) get into conflict, (might) behave, etc. It will be very useful for them to query a system for Semantic-supported management of Web Services that integrates aspects from multiple WS* descriptions — which has not been possible so far, but is now allowed by the approach we present here and system we have briefly introduced in section 2.

Question 2 What does he/she/it use the semantic descriptions of Web Services for?

There are numerous use cases where the integration of semantic descriptions may help the developer or administrator. Hence, the list below is neither exhaustive nor are the individual use cases mutually exclusive. The reader may note that it is germane to semantic descriptions to state what there is and not how it is to be combined and what is its sole purpose.

Question 3 When does he/she/it use the semantic descriptions of Web Services?

We consider three different stages, viz. development time, deployment time and runtime.

Question 4 Which aspects should be formalized by our ontology?

On one hand, we want to be able to automate management tasks covering a broad range of aspects (like security, policies, interface descriptions etc.). On the other hand, the complexity of the ontology has to be kept small to avoid overburdening the developer. The use cases serve as modelling requirements also for building a suitable management based on an ontology like in section 1.1.

3.1 Selecting Service Functionality

As more and more services become available, developers will need some tool support in browsing and selecting an appropriate service. The canonical approach to this task is a taxonomic categorization of services according to their capabilities. Naturally, searching for services of a certain capability class C should also yield all services classified as instances of subclasses of C.

Who:	Developer
What for:	Service discovery and selection
When:	Development time
Which aspects:	Service taxonomy

3.2 Incompatible Inputs and Outputs

Type checking is not as straightforward anymore, using loosely coupled services operated by a large number of organisations. Furthermore, the interpretation of a B2B term such as ‘price’ might be different, even though syntactically it refers to an agreed-upon XML Schema type. For instance, different partners might have different assumptions about the currency and taxation details. A system, which automatically compares communication inputs and outputs according to a more detailed model, will help to prevent unexpected behaviour in a system.

Note that a developer who uses Web Services wants to check at *development* time whether some incompatible configuration exists. While a 100% solution, such as required for full automation, will remain unfeasible in most cases, ticking of 80% of problematic situations by semantic support is a very desirable feature of semantic management.

Who:	Developer
What for:	Code debugging
When:	Development time
Which aspects:	Interface description

3.3 Detecting Loops in the Invocation Chain

Web Services based applications usually make use of asynchronous messaging, bringing upon quite complex interaction protocols between business partners. Current workflow design workbenches only visualize the local flow and leave the orchestration of messages with the business partners up to the developer. We believe that enough information is available in machine-readable format such that a tool can assist the developer in this task. For instance, the structure of the local flow can be combined with publicly available abstract flows of the partners in order to detect loops in the invocation chain that would lead to non-termination of the system.

As shown in the bioinformatics domain [LBWS04], automated composition of workflows is likely to be inappropriate in most cases. Hence, we propose to support the developers in their management tasks and not to replace them.

Who:	Developer
What for:	Code debugging
When:	Development time
Which aspects:	Workflow information (plans)

3.4 Relating Communication Parameters

This use case is again motivated by e-business policies. Assume a policy states that the entire supply chain must consist of ISO 9000 certified partners. Enforcing this policy requires correlating communication paths with information about the organizations operating the communication endpoints. Another example would be a policy stating that confidential information should only be sent across a secure communication channel. In this case, knowledge about message payload types such as credit card information must be connected with the properties of the underlying transport.

Who:	Developer, Administrator
What for:	Plausibility check on flow
When:	Development and deployment time
Which aspects:	Service profile

3.5 Policy Matching

Policies play an increasing role, as demonstrated by the recent WS-Policy proposal. The idea of a policy is to lay out general rules and principles for service selection. Thus, rather than deciding whether an invocation is allowed on a case by case basis at runtime, one excludes services whose policy violates the local policy at development time. The major benefit is that policies can be specified declaratively. Since policies can change dynamically, one can also imagine a scenario, where a service bus matches policies during runtime in order to select an appropriate provider out of a pool of competitors. At this point, the reader may note, that the current policy-related proposals are not based on a uniform logical framework. We believe that this would make the realization of a policy engine and the integration of the various approaches from the areas of privacy, security, or access control much easier.

Who:	Developer, System
What for:	Excluding unsuitable services
When:	Development and runtime
Which aspects:	Policies

3.6 Analyzing Message Contexts

Message passing plays the central role for Web Services. A message sent to a component can in turn trigger several other messages being sent out on behalf of the initial message. Messages may carry a context with information about the sender, the sender's credentials, or the message's transactional context. During the deployment of a component or flow, the administrator makes important choices as to how messages are propagated. These include whether the sender information is carried along or whether the new message is sent on behalf of a new user (also called the 'run as' paradigm). Similar choices are made with respect to the transactional settings. Components and flows can choose to always open a new transaction, require a prior transactional context, or open a new transaction when needed. In a large network of direct and indirect invocations, it is crucial to be able to detect configuration errors, such as a situation where a component switching to user context X and calling Y, which does not have user X in its access control list.

Who:	Administrator
What for:	Plausibility check on deployment settings
When:	Deployment time
Which aspects:	Service profile

3.7 Aggregating Service Information

Services will often be implemented based on other services. A service provider publishes information about its service. This might include service level agreements indicating a guaranteed worst-case response time, the cost of the service, or average availability numbers. The service requestor, in this case a composite service under development, can collect this information from the respective service providers. In turn, it offers a service and needs to publish similar numbers. We envision a tool that supports the administrator by providing a first cut of this data by aggregating the data gathered from external providers.

Who:	Administrator
What for:	Suggestion for deployment parameters
When:	Deployment time
Which aspects:	Quality of service

3.8 Quality of Service

While the previous use case was based on data gathered from service providers, one might want to obtain own statistics on the reliability and availability of business partners' IT infrastructure. Assuming the system is aware of potential endpoints implementing a required service, these endpoints can be pinged regularly. If an actual request arrives, aggregated availability information can be used to direct subsequent requests to one or the other third party service. Likewise, a provider needs to make sure it provides an adequate service level for its customers. In case of performance bottlenecks, it might have to make an educated decision on which jobs to grant higher priority and which job to drop or decline. Existing service level agreements and of course the respective penalties play an important role here.

Who:	Administrator, System
What for:	Performance optimization
When:	Runtime
Which aspects:	Quality of service

3.9 Overview of the Use Cases in Context

Figure 2 depicts the described use cases for semantic modelling in the context of the overall development process for one application. Depending on the application requirements and the organisational characteristics some or all of the use cases can be supported by semantic-supported management for Web Services. The separation gives more flexibility to start with the most promising tasks for Semantic-supported management while still manually dealing with some more specific cases.

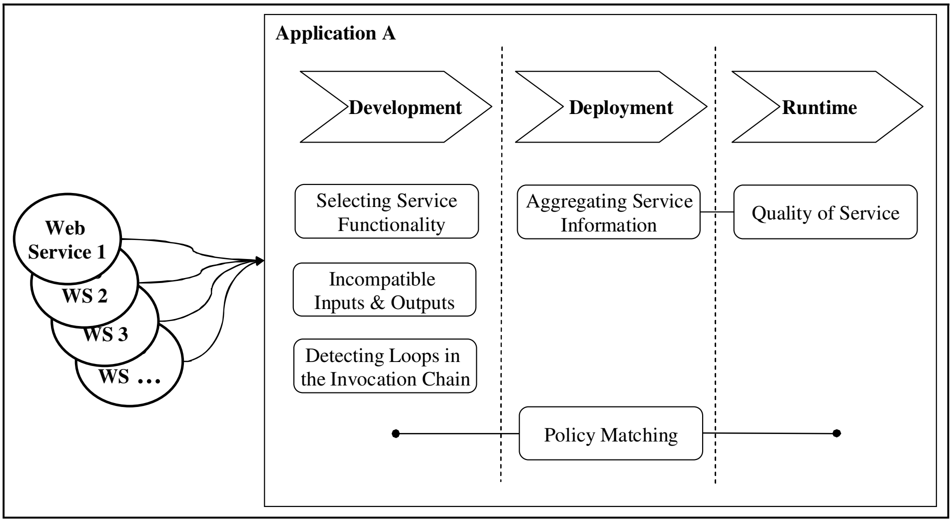

Figure 2: Overview of the Use Cases for Semantic Modelling in WS-Development and Management

4 Economic Analysis of Semantic Modelling for Web Services Management in Organisational Contexts

The economic analysis of the processes and effects related to semantic modelling reveals different kinds of impacts. Some of them are directly influencing the results. Others at first glance only indirectly contribute to the process but are overall much more influential. They mostly are long-term effects. This obstacle often impedes the cost-benefit analysis of new technologies. [KA89] For analysing the effects of different options of Web Services management the following costs must be taken into account:

- working costs (for development and administration)
- investments (in tools and semantic models)

The benefits also require careful attention. Benefits will differ due to the quality of the Web Services provided. Especially the number of errors and their resolution influence the performance and they will depend heavily on the management of the Web Services.

The process of WS-management is based on different kinds of interrelated information. The processes of the three options discussed are using and transforming these information differently to support the underlying tasks (see Figure 3). For WS* manual Management the developers use the available information directly to specifiy the XML-documents covering the different concerns of WS*-Management like Process flows, Security policy or Interface definitions. They are used as parameters for running the Web Services. Semantic-supported and automatic semantic management on the other hand start with modelling the requirements and its constraints in a semantic model based on an underlying ontology. Automatic semantic management takes these models directly for configuring the Web Services. In the Semantic–supported Management the developers must translate the semantic models into technical formats (here Parm.-Doc which usually are XML). As modelling is on a fairly moderate level there are some tools available which support the creation of semantic models by automatically analysing existing Web Services information. [Ob06]

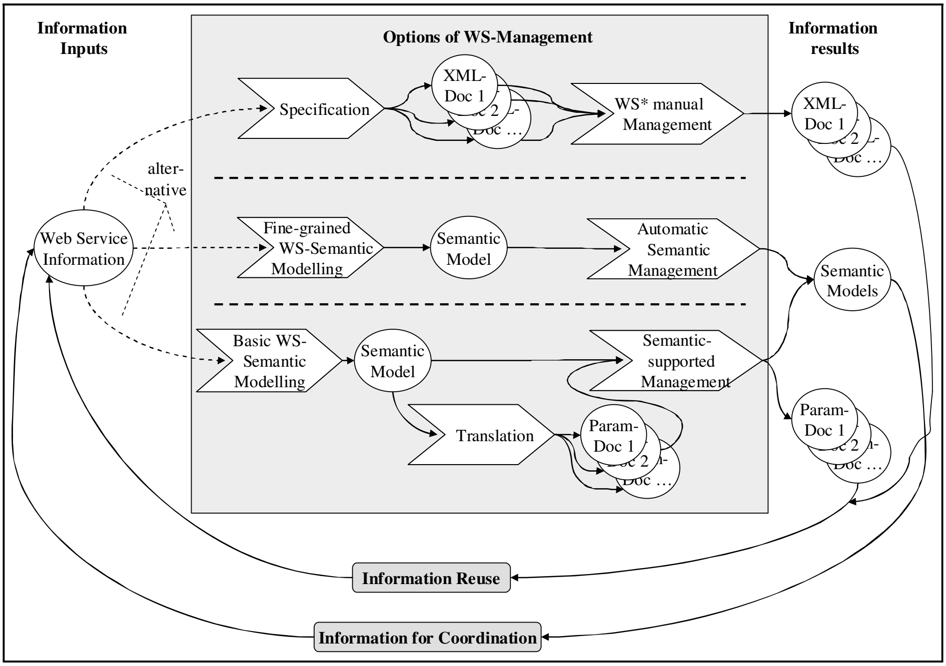

Figure 3: Necessary Information for Development and Management of Web Services

All three process options for Web Services management deliver results which can be reused in developments for similar Web Services. This will result in some cost savings due to the individual learning curve in the development of Web Services. On the other hand the necessary effort will rise if a considerable number of applications based on Web Services have been introduced. Then the effort for coordination of the interdependencies rises substantially [St00] and integrated information on the Web Service-based applications will be crucial. WS* manual Management will become quite tedious while the coordination of considerable numbers of Web services will be well supported by the semantic models for Semantic-supported and Automatic semantic Management.

Comparing the information handling in the three processes for WS-Management the first and the second process are simpler than the third Semantic-supported process.
1. Individually taken the first process of WS* manual Management will be most cost effective.
2. The process of WS-semantic modelling for Automatic semantic Management seems quite straightforward, but as this kind of modelling of Web Services must be very fine grained it will incur high costs. The semantic modelling will also require some investment for the basic ontology. This investment will depend on the standards available and the requirements of the application. So a sufficiently high number of Web Services with their interdependencies is necessary to compensate the investment in an

initial ontology and the essential tools.
3. In the case of the Basic-Semantic Modelling for Semantic-supported Management the additional effort for semantic models must be repaid by the savings due to an improved coordination of interdependencies, too. But the investment and the modelling costs will be lower due to the more pragmatic level of modelling and the tools available.

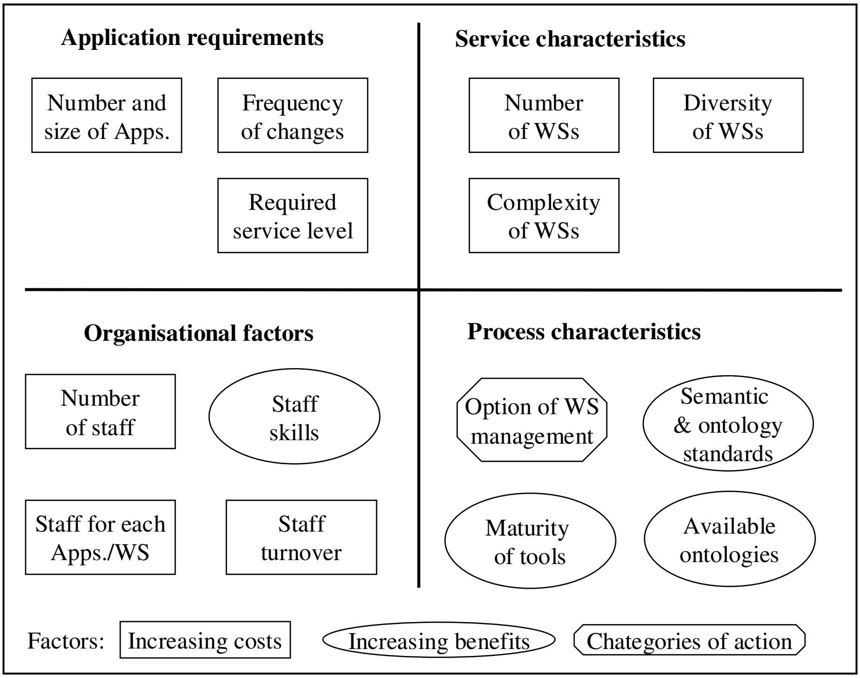

Figure 4: Main influencing factors for costs and benefits of semantic modelling for Web Service Management

As it has been mentioned before, the break-even for the different options does not only depend on technical characteristics. The analysis must include crucial factors from different areas: the originating *Application requirements*, the *Organisational factors*, *Service* and *Process characteristics*. Figure 4 contains the four areas with major factors. The factors noted in rectangles are increasing costs. This means an increase in their value effects an increase in necessary efforts. Other factors that decrease the individual effort like the availability of appropriate tools are depicted in an ellipse. The option of WS-management is indicated by an octagon. It stands for the different categories of action with subtle effects on costs and benefits.

5 Comparison of the Main Characteristics of the WS-Management Options

In the preceding sections a number of different characteristics of the process of WS-management have been identified. Most important are the information necessary, the factors that determine the business, the organisational and the technical complexity. Depending on the circumstances a different option of WS-management is often advisable. WS* with manual management activities is likely to be best choice for single applications in a small scale WS environment because it directly handles all matters on the technical platform and does not entail any additional preparative work. Semantic management on the other hand requires some investment which will only repay if a considerable number of WS-based applications are developed or dedicated requirements can only be met by higher formalization with semantic models.

WS-Management Options	Necessary additional activities	Limiting Factors	Supporting Factors	Main Weakness	Main Advantage
WS* manual Management	--	Overall number and complexity of WSs	Small number and complexity of WSs	Quality assurance difficult	No investment necessary
Semantic-supported Management	Basic semantic modelling	High quality requirements	Active conceptual modelling	Some initial costs	Flexible for changes
Automatic semantic Management	Fine grained semantic modelling	High initial effort	Standard ontologies	High initial costs	High quality management of complex WSs

Table 1: Limits, facilitators, strengths and weaknesses of the WS-Management options

For being able to judge future developments it is essential to identify the drivers and inhibitors for different kinds of WS-management. Table 1 lists main limiting and supporting factors for each option of WS-management. Semantic supported management is positioned in-between the others because it combines a medium capacity to handle complexity with some but not high costs. The supportive factors may act as very potent facilitators if either the common practice in the company is based on conceptual modelling or if evolving standards and repositories minimize the effort for semantic management activities.

5.1 Development of the Usage of WS-Management Options

Web Services have been considered as cure for highly integrated but inflexible software systems. [TG05] As they can be backed by computers ranging from small dedicated systems to big legacy applications they offer at least theoretically all kinds of computing services. Anyway, until now the diffusion of Web Services has not been very substantial. One reason is that the stability and reliability which has been reached by the integration in current systems is in most cases also necessary for WS-based applications. This reliability can only be attained in Web Service-based applications by a careful management of numerous features. [JoRa05] Manually done this is quite a tedious exercise. The suggested solution of automatically managing Web Services by ontologies requires a very high investment. It is also not clear whether an automated management is feasible at the moment because of many open questions and just emerging standards.

So on one hand, many concepts for Web Services Management are available and there is quite some demand for Web Service-based applications but, on the other hand, a number of practical obstacles remain for their more extensive usage. With the approach of semantic-supported management we advance a pragmatic and evolutionary concept for employing Web Services. Semantic-supported management relies on semantic models but uses them selectively depending on their appropriateness for informative or semi-automatic purposes in the management of Web Services. One main advantage of Semantic-supported management is that the resulting ontology provides the groundwork for later automation. Depending on the development of a number of other variables discussed the distribution of WS-Management options in an organisation may result in a curve similar to Figure 5.

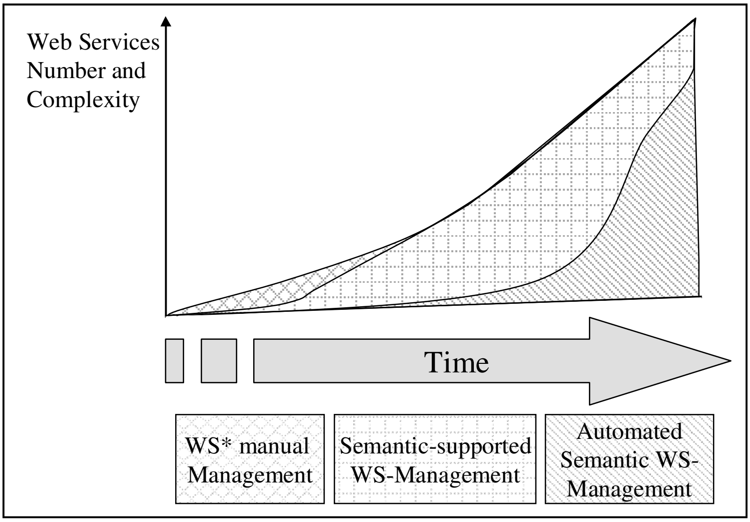

Figure 5: Probable development of the usage of different options for Web Service Management

6 Conclusion

Web Service management is of crucial importance for the vision of Web Service-based applications as an infrastructure for flexible business processes. Anyway, the common usage of Web Services will not happen overnight. This is due to many reasons which are presented in this paper combining technical, organisational and economical research. The central point addressed is the difficulty to handle the information required for decisions in Web Service management. With this focus we analysed the process of Web Services management. As a result, the overall Web Service management activities have been separated into a number of use cases. Also a number of decisive factors and circumstances have been identified. These factors apply to all aspects of the Web Services management while the use cases help in a more detailed reflection for judging specific facets. As the usage of Web Services is supposed to increase the decision on the kind of WS-management is dependent on time and circumstances. So the presented elements of an analysis can also serve to determine beforehand potential problems and smooth the way for an effective dissemination of Web Service-based applications.

References

[EDBS05] Ebert, C.; Dumke, R.; Bundschuh, M.; Schmietendorf, A.: Best Practices in Software Measurement. Springer, Heidelberg, 2005.

[GBC04] Gangemi, A.; Borgo, S.; Catenacci, C.; Lehmann, J.: Task taxonomies for knowledge content. Metokis deliverable d07, Jun 2004.

[GST04] Gangemi, A.; Sagri, M.-T.; Tiscornia, D.: A Constructive Framework for Legal Ontologies. Internal project report, EU 6FP METOKIS Project, Deliverable, 2004.

[JoRa05] Jong W.K.; Radhika J.: Web Services Composition with Traceability Centered on Dependency. In: 38 th HICSS, Hawai, 2005; pp. 1-10.

[KA89] Kaplan, R. S.; Atkinson, A. A.: Advanced Management Accounting. 2nd ed., Prentice-Hall Int., Englewood Cliffs, NJ, 1989.

[LBWS04] Lord, P.; Bechhofer, S.; Wilkinson, M. D.; Schiltz, G.; Gessler, D.; Hull, D.; Goble, C.; Stein L.: Applying Semantic Web Services to Bioinformatics. In: 3rd Int. Semantic Web Conference, volume 3298 of LNCS. Springer, 2004; pp. 350-364.

[Ma04] Maier, R.: Knowledge Management Systems. 2.nd ed.: Springer, Heidelberg, 2004.

[MSZ01] McIlraith, S. A.; Son, T. C.; Zeng, H.: Semantic Web Services. IEEE Intelligent Systems, 16(2), Mar 2001; pp. 46–53.

[Ob06] Oberle, D.: Semantic Management of Middleware. In: The Semantic Web and Beyond, vol. 1. Springer, New York (to appear) 2006.

[OLES05] Oberle, D.; Lamparter, S.; Eberhart, A.; Staab, S.: Semantic Management of Web Services. In: Proc. 3rd International Conference on Service-Oriented Computing, Amsterdam, 2005.

[OLGV05] Oberle, D.; Lamparter, S.; Grimm, S. Vrandecic, D.; Staab, S.; Gangemi, A.: Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems. Technical Report, University of Karlsruhe, AIFB, 2005.

[OGG02] Oltramari, A.; Gangemi, A.; Guarino N.; Masolo, C.: Sweetening ontologies with DOLCE. In: Ontologies and the Semantic Web, 13th Int. Conference, EKAW 2002, Proceedings, 2002; pp. 166-181.

[SS04] Staab, S.; Studer, R.: Handbook on Ontologies. International Handbooks on Information Systems. Springer, Heidelberg, 2004.

[St00] Sterman, J. D.: Business Dynamics - Systems Thinking and Modelling for a Complex World. McGraw-Hill, Boston, 2000.

[TG05] Tamm, G.; Günther, O.: Webbasierte Dienste: Technologien, Märkte und Geschäftsmodelle. Physica, Heidelberg, 2005.

[WF05] Wolff, F.; Frank, U.: A Multiperspective Framework for Evaluating Conceptual Models in Organisational Change. In (Bartmann, D. et al Eds.): 13 th European Conference on Information Systems (ECIS), Regensburg, AIS, 2005; pp. 1-12.