Using Thematic Grids to Document Web Service Operations

Jan Christian Krause
AKRA GmbH, Domstraße 17, 20095 Hamburg, Germany
jan-christian.krause@akra.de

Abstract

Web Services are frequently used for system integration in global contexts. Therefore their documentation is required to be precise and complete. This paper discusses state-of-the-art approaches to document Web Services and provides examples of the negative impact a lack of construction guidelines has on the documentation of what the service does. A verb-focused approach to documentation construction is presented which is based on the linguistic concept of thematic roles and grids. It is complemented by an empirical study, showing that the prerequisites of the approach are satisfied. The presented findings contribute to ongoing research in the area of semantic Web Service documentation.

1. Introduction

Software documentation has a long tradition in software engineering. In recent years a lot of research has been done to increase precision and quality of software documentation through standardized notations (e.g. UML, SysML, javadoc, etc.) and tools supporting these notations on different granularity levels (e.g. Borland Together, the Enterprise Architect from Sparxsystems, etc.). Since it describes the interaction between systems, the documentation of software interfaces plays an important role for system-integration in a global context (e.g. between different departments, companies or governments). In practice, the concept of a Web Service is widely supported and therefore a frequently used approach to integrate systems.

To design, implement and test the interaction with a Web Service with minimum effort, its documentation is required to be precise and complete. In line with the principles of information hiding and encapsulation, the documentation should only describe effects which are visible to the consumer [1]. The completeness requirement leads to the question, whether a set of parameters and affected resources, as well as their role for a given operation (e.g. a parameter is used as filter criterion or as a location where to search, etc.) could be derived from the Web Service's interface. In most cases this information cannot easily be extracted from the operation's signature, because in many practical implementations a Web Service is initialized with external resources which it does not receive as parameters at the invocation time of an operation, e.g. the connection to a database. This paper investigates this problem using the example of SOAP-Web Services with interfaces defined by means of the Web Service Description Language (WSDL).

2. Related work

The Web Service Description Language, standardized by the World Wide Web Consortium (W3C), is used to define the interfaces (WSDL 1.1-section: *porttype*, WSDL 2.0-section: *interface*), the deployment locations (WSDL 1.1-sections: *services* and *ports*, WSDL 2.0-sections: *services* and *endpoints*) and the message exchange formats of Web Services (WSDL 1.1 and 2.0-sections: *types* and *messages*) [2,3]. The WSDL provides the element *documentation,* which is allowed as a sub-element of all other WSDL-elements, but it does not specify any attributes or sub-elements of the *documentation*-element. It merely says that the *documentation*-element is a container for human-readable or machine-processable documentation without any guidelines for its appropriate use. The grid of elements to for document each operation can easily be derived from the declared WSDL-elements of the operation. The W3C's web services architecture calls this kind of Web Service documentation syntactic functional description [4].

Two other initiatives work under the hood of the W3C on the standardization of semantic service descriptions called WSDL-S or OWL-S [5,6]. WSDL-S is an extension of WSDL which covers pre- and post-conditions (effects) of operations, Service Categories and Model References. Service Categories provide the syntax to categorize services. A Model Reference is an attribute of a WSDL-element containing a reference to an external semantic model or ontology. In OWL-S services are described by a service profile, service grounding and a service model. The service profile describes the input and output of an operation. It also contains preconditions and a service category. The service grounding describes how to access the service. The service model explains how the service works by means of a special language for the program flow which consists of statements, conditions, loops, etc.

While the definition of pre- and post-conditions follows the approach of axiomatic semantics, the definition of a service model adopts the approach of operational semantics from computer science. None of the presented approaches supports the direct derivation of the required resources from the interface description. One could argue that this information could be extracted from the OWL-S service model, but this would only delegate the task of documentation to the Atomic Processes of OWL-S. If the required parameters and resources of an Atomic Process are not documented, the parameters and resources of the Composite Process or Web Service using it could not be derived. Furthermore, this approach requires deep knowledge of how the Web Service works. It results in an additional modeling effort for the creation of the service model.

In recent years many researchers and practitioners have started to develop service ontologies for the description of Web Services [7]. These ontologies focus on the integration of different aspects into a service description, e.g. policies and access restrictions, pricing, general terms and conditions of use, etc. But none of them concentrates on the derivation of required parameters and resources.

Since current documentation approaches for Web Services derive the set of elements to be described (parameters, resources, exceptions and results) only from what has been explicitly declared in the signature of the corresponding operation, there is a need for to also identify possible hidden elements from this description.

3. Using thematic grids to derive parameters and resources of an operation

In natural language, activities are mainly described by means of verbs, while the actors or attributes of the activity depend syntactically and semantically on the verb. For this reason it is common to choose the identifier of a Web Service operation in a way that it contains a verb which describes the activity to be carried out. A simplified example is given in Table 1.

Name of Operation:	findCustomerById
Contents of Output-Message:	customer::Customer
Contents of Input-Message:	id::integer
Faults:	None

Table 1. Example for the signature of a Web Service Operation

From a linguistic point of view, this labeling style is an imperative sentence (with a missing article, a missing conjunction and two missing personal pronouns: Find *the* customer by *his or her* id.). In linguistics, the concept of phrases is used to structure natural language sentences [8]. Phrases split sentences into typed groups of words, which can be moved or replaced without rendering the sentence ungrammatical. Each phrase consists of a phrase-head (the semantic core of the phrase) and a tail of dependent phrases. A phrase could contain one (minimal phrase) or more words (complex phrase). The phrase structure of a sentence could be visualized by a phrase-structure-tree.

For example, the verb *find* binds two other nominal phrases (arguments of the verb): the subject and the object. The formalism used to specify the arguments of a verb is called a thematic grid and each required argument is called a thematic role [9,10,11]. An argument could be mandatory or optional. A simplified thematic grid for the example verb *find* would be

Find [OBJECT] [COMPARISON] [SOURCE].

Here the thematic roles object, comparison and source are used. The role agent is only implicitly given (“you”). Its entity initiates and carries out the activity described by the verb. The object is the entity which is affected by the action [12]. The comparison provides a criterion which classifies the object and the source specifies the place where the object comes from [13]. In this example, the agent and object are mandatory arguments, while the comparison and source are optional. Now the operation’s signature could be matched to the thematic grid, for example as

Find [$_{\text{OBJECT}}$ Customer] [$_{\text{COMPARISON}}$ id] [$_{\text{SOURCE}}$].

In this example the source is not specified as a parameter. Nevertheless the algorithm has to look somewhere for the customer with the given id. The necessity of including this information into the documentation is contributed by the thematic grid of the verb. By assigning its thematic roles to the elements mentioned in the operation’s signature, a consumed storage-resource (e.g. a database) has been identified, which is not mentioned in the operation’s signature. In line with [1] this resource should definitely be documented because the result of the operation depends crucially on the availability and the state of this resource. As described in section 2 current state-of-the-art approaches to the documentation of Web Services derive the set of elements to be documented only from the operation’s signature and therefore would not cover the source.

By using the thematic grid of the verb used in the operation’s identifier and assigning the signature-elements to the thematic roles of this grid, required hidden parameters and resources can be identified without deep knowledge about the implementation of the operation. Furthermore the use of thematic roles can decrease the effort for documentation, because the developer only assigns the uninstanciated role of the grid to the corresponding parameter or resource. It is assumed that in many cases the thematic role explains the parameter sufficiently. Possibly this advantage will lead to a high degree of acceptance on the side of developers. If necessary, the documentation could also be complemented with pre- or post-conditions or textual annotations.

4. The use of verbs in identifiers of Web Service Operations

The derivation of thematic grids depends on the existence of a verb in the identifier of the corresponding Web Service operation. This leads to question Q1.

Q1. Do identifiers of Web Service operations usually contain a verb?

To avoid the necessity to define thematic grids individually for each verb, similar verbs can be grouped into classes which share the same thematic grid. Therefore an appropriate verb-class-system has to be identified, which covers as many as possible verbs actually occurring in identifiers of Web Service operations as is expressed by question Q2.

Q2. Which verb-class-system covers the maximum number of the verbs used in identifiers of Web Service operations?

To answer both questions a representative corpus of Web Service interfaces is required. The Web Service – search engine seekda GmbH provided a snapshot of its Web Service-index to be used in this investigation, which is described in detail in section 4.1.

Web Services are frequently used for workflow automation in business contexts. It is common practice of software-development in business contexts to formulate identifiers in English. Therefore a verb class system appropriate for these contexts is required. Two class systems have been identified to fit this requirement [14]: the MIT Process Handbook [15] and the verb classes proposed by Levin [16]. These verb class systems were introduced in sections 4.2 and 4.3.

4.1. The seekda-Corpus

The company seekda GmbH, located in Innsbruck, Austria, offers an online search engine for Web Services[1]. Its index contains interface-descriptions for Web Services in WSDL-notation[2]. seekda uses robots to crawl the web for WSDL-descriptions and index them automatically.

Corpus-Attribute	Attribute-Value
Countries of service-providers[3]	89
Number of service-providers	8,947
Number of WSDL-descriptions	17,453
Number of PortTypes (Interfaces)	24,471
Number of Operations	186,741

Table 2. Quantified attributes of the seekda-corpus

Some quantitative details of the corpus are given in Table 2. As the seekda-corpus contains a high number of operations from a wide range of Web Services, it is assumed to be a representative snapshot for a practical Web Service system-integration environment.

4.2. The MIT Process Handbook

The development of the MIT Process Handbook started in 1991 with the goal of creating an online library for sharing business knowledge. Today it contains more than 5,000 business activities related to each other via an inheritance hierarchy [15]. All activities are formulated in English. The root of the hierarchy is the verb *act* which is specialized by eight generic verbs: *create*, *modify*, *preserve, destroy*, *combine*, *separate*, *decide* and *manage*. These verbs have been derived from the lexical database WordNet which today contains over 25,000 verbs [17]. One verb can be assigned to more than one class.

4.3. The Extended Levin Classes

In 1993 Levin proposed a verb class system which covers more than 3,000 English verbs. In contrast to the MIT Process Handbook Levin's work is based on linguistic and not on empirical analysis [16]. This system is frequently used for NLP-applications. Korhonen and Briscoe extended the Levin classes by 57 new classes and 106 diathesis alternations for verbs [18]. To achieve the maximum verb coverage in this investigation the Extended Levin Classes proposed by Korhonen and Briscoe and implemented in the VerbNet 3.0 lexicon have been used [19]. As with the MIT Process Handbook, in the Extended Levin Classes one verb can also be classified in more than one class.

4.4. Data analysis

The first step of the analysis was to extract the used verbs from the corpus according to the following procedure:

[1] For detailed information about the company seekda GmbH and its searchengine visit http://seekda.com/.

[2] The cache-snapshot is dated to 11th September 2009.

[3] Derived from the providers internet top-level-domain (e.g. .de, .fr, etc.)

1. All operation identifiers have been extracted from the corpus by using a SAX-parser (event-driven parser for XML). An operation identifier is the value of the *name*-attribute of the *wsdl:operation*-element (in scope of *wsdl:porttype*).
2. Since the lower Camel Case-convention is used in most cases, the identifiers have been split into tokens at each capital letter.
3. Each token's part-of-speech has been determined by tagging the token with a Part-Of-Speech-Tagger. For that purpose the Stanford Tagger 1.6 with the PennTreebank-Tagset has been used [20].
4. All tokens tagged as infinitive or present tense verb (tags VB, VBP and VBZ) have been collected.
5. Each verb candidate has been checked for really being a verb by a look-up in the WordNet database. All wrongly tagged tokens have been removed from the verb-statistics.

This procedure delivers the set of verbs used in the corpus and their frequencies can be used to answer question Q1. For question Q2 these verbs have to be classified according to the MIT Process Handbook and the Extended Levin Classes. A class is considered to match a verb if the verb or one of its synonyms is member of the class. Again, the synonyms have been determined by means of WordNet. With this information it was possible to compute which verb class system covers more operations and more verbs.

For some tokens it depends on the grammatical context whether they have to be tagged as a verb or as a noun. The token *email* for example, is a verb in identifier *email invoice*, but a noun in the identifier *check email*. To obtain a representative snapshot of English verbs actually used in the Web Service-domain this ambiguity has been reduced by combining the Stanford Tagger with WordNet information. All tokens with the tags VB, VBP and VBZ are cross checked with WordNet. This ensures that only verbs with the grammatical role of a verb are considered. To compare the two verb class systems the tagged identifiers are classified into the four categories described by Table 3.

Since the Stanford Tagger follows a probabilistic approach and has been trained on sentences with a typical subject-predicate-object-structure, a high number of Not Classified Identifiers can be expected. Take the identifier *Process XML* as an example. The tagger would tag it as *Process_NN XML_NNP*. Here the token *Process* has been wrongly tagged as a common noun but *XML* correctly as a proper noun. This result could be explained by the structure of the training sentences, which start in most cases with the subject. Since the subject is often a noun or a personal pronoun, the probability for such a tag is relatively high. To convert the Not Classified Identifiers into a form which resembles the training data more closely, the token "We" has been added before each Not Classified Identifier, e.g. *We Process XML*. The Stanford Tagger will tag this sentence now as *We_PRP Process_VBZ XML_NNP* which is correct. The analysis procedure described above is applied again to the converted Not Classified Identifiers. Finally, the verb frequency statistics of both runs have been merged and the verb class system classification ratios have been computed.

Identifier Category	Contains a verb	Verb is tagged
Considered Identifier	Yes	Yes
Not classified identifier	Yes	No
Wrongly classified identifier	No	Yes
Correctly ignored identifier	No	No

Table 3. Systematic to handle tagged identifiers

5. Results and discussion

The corpus contains 161 identifiers with a dot, e.g. *clients.updateClient*. These identifiers are interpreted as two sentences by the tagger and therefore not considered in the investigation. This results in a corpus consisting 186,580 identifiers. Table 4 shows their distribution on the unmodified data.

Identifier Category	No. of class. identifiers
Considered Identifiers	111,488
Not classified identifiers	59,038
Wrongly classified identifiers	510
Correctly ignored identifiers	15,544
∑	186,580

Table 4. Identifier classifications after the first run

The second run of the experiment was based on the 59,038 Not Classified Identifiers, which have been extended with the personal pronoun "We" at their beginning. Due to this modification the tagger was able to tag the verbs correctly in 38,700 cases. 7,158 identifiers were wrongly classified by the tagger and 13,180 identifiers remain as not classified. The overall results of both runs are listed in Table 5.

Identifier Category	No. of class. identifiers
Considered Identifiers	150,188
Not classified identifiers	13,180
Wrongly classified identifiers	7,668
Correctly ignored identifiers	15,544
∑	186,580

Table 5. Identifier classifications after the second run

830 verbs have been extracted from the 150,188 Considered Identifiers. The 30 most frequent verbs are listed in Table 6. It is interesting to note the distribution of verb frequencies which resembles an exponential decrease. The most frequent verb *get* occurs nearly 12.2 times more often as the verb *add* on the second rank, but *add* occurs only 1.2 times as often as the third ranked verb *create*. It seems that Zipf's law applies for this verb-distribution, which states that the product of rank and frequency of a word is constant [21]. This observation could be explained by the very general meaning of *get* which fits for every operation that returns something. The usage of *get* in an identifier decreases the effort for the developers to choose a more precise verb. This corresponds to another observation, namely that the 30 most frequent verbs have very technical characteristics and are often used in programming languages, such as *get* (1), *run* (13) or *do* (20), in query languages, such as *delete* (4), *update* (5) or *describe* (30) or as console command names, such as *find* (14), *echo* (15), *edit* (24) or *test* (29). Since in a technical context these verbs refer to implementations and these implementations have interfaces with operations, the verbs are also associated with the operations, their results and parameters. It could be assumed that a developer of a Web Service chooses such a verb because of the high correlation between the signature of the operation and the general thematic grid of the verb in this technical context. This lends further support to the approach described here, which derives general parameters and resources of an operation from the thematic grid of the verb. In general $\frac{150{,}188 + 13{,}180}{186{,}580} \approx 0.88$ of the identifiers contain an English verb, which leads to the approval of question Q1.

Because the number of Not Classified Identifiers and Correctly Ignored Identifiers are relatively high, the identifiers of these categories have been reviewed manually. It turned out that the Correctly Ignored Identifiers fall into four categories:

- The identifier is not formulated in English.
- The identifier does not contain a verb, such as *optional info* or *from csv to xml*.
- The identifier does not follow the Camel Case Syntax, e.g. *getserverversion*.
- The verb of the identifier has been abbreviated, e.g. *mkdir* (maybe "make directory") or *crmget inbox folder* (maybe "get inbox folder of customer relationship management system").

The Not Classified identifiers were mainly valid imperative sentences, which have been incorrectly tagged by the Stanford Tagger even though the personal pronoun "We" has been added to the identifier, such as *We message query data* (tagged as: We_PRP message_NN query_NN data_NNS) or *We order* (tagged as: We_PRP order_NN). It is assumed that the number of Not Classified Identifiers could be further decreased by training the tagger on a special corpus of imperative sentences.

Rank	Verb	Occurrences
1.	get	74,790
2.	add	6,105
3.	create	4,929
4.	delete	4,769
5.	update	4,398
6.	send	3,176
7.	list	2,658
8.	check	1,944
9.	remove	1,861
10.	is	1,845
11.	load	1,762
12.	search	1,742
13.	run	1,727
14.	find	1,367
15.	echo	1,146
16.	submit	1,041
17.	save	1,018
18.	wait	949
19.	modify	944
20.	do	913
21.	validate	841
22.	register	836
23.	insert	728
24.	edit	701
25.	query	553
26.	retrieve	539
27.	process	535
28.	change	526
29.	test	524
30.	describe	500

Table 6. The 30 most frequent verbs used in the seekda-Corpus

	Class. identif.	Not class. Identif.
MIT Process HB	147,454	2,734
Ext. Levin Class.	147,890	2,298

Table 7. Identifier classification of both verb class systems

	Class. verbs	Not class. verbs
MIT Process HB	644	186
Ext. Levin Class.	744	86

Table 8. Verb classification of both verb class systems

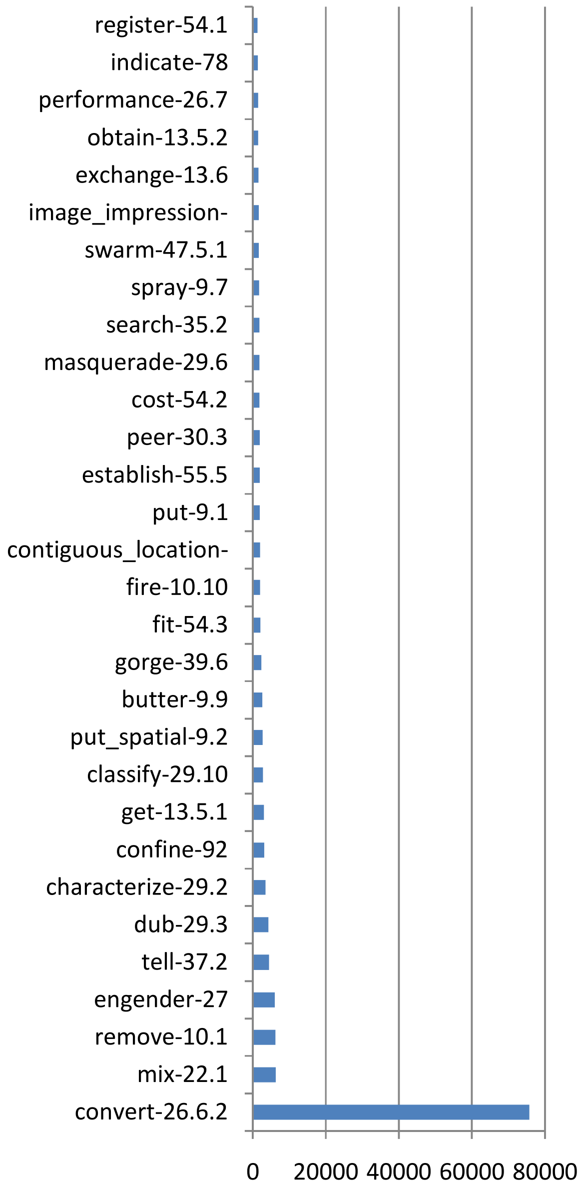

Figure 1. Distribution of Considered Identifiers in the first 30 Extended Levin Classes

Table 7 and Table 8 show the results of comparing the MIT Process Handbook with the Extended Levin Classes. The number of classified identifiers is nearly the same, which results in a coverage ratio of $\frac{147{,}454}{150{,}188} \approx 0.982$ for the MIT Process Handbook and of $\frac{147{,}890}{150{,}188} \approx 0.984$ for the Extended Levin Classes. However there is a large difference between the two verb class systems with respect to their coverage: the MIT Process Handbook covers $\frac{644}{830} \approx 0.78$ while the Extended Levin Classes cover $\frac{744}{830} \approx 0.90$ of the verbs. Both observations can be explained by the exponential fall of the frequencies in the verb distribution. 489 of all verbs occur 10 or less times in the corpus. So their impact on the number of classified identifiers is rather small, but significant for the number of verbs. Due to these observations and with question Q2 in mind the Extended Levin Classes should be preferred over the MIT Process Handbook.

Figure 1 and Figure 2 show the distribution of the Considered Identifiers for both verb class systems: the MIT Process Handbook and the Extended Levin Classes. Since one Considered Identifier can be classified into more than one class, the degree of this ambiguity could provide an additional selection for choosing a particular class system. As the figures show, the MIT Process Handbook has a much higher degree of ambiguity than the Extended Levin Classes. This seems plausible because the number of classes in the Levin-System is much higher compared to the MIT Process Handbook. But as the distribution shows, the Extended Levin Classes are really more useful to separate the identifiers from each other. This is very important, because the thematic roles are defined per verb class. On average the Extended Levin Classes classify each identifier into 1.25 classes, while the MIT Process Handbook returns 3.38 classifications per identifier. This finding also supports the preference for the Extended Levin Classes, because here the thematic roles could be defined specifically for each class.

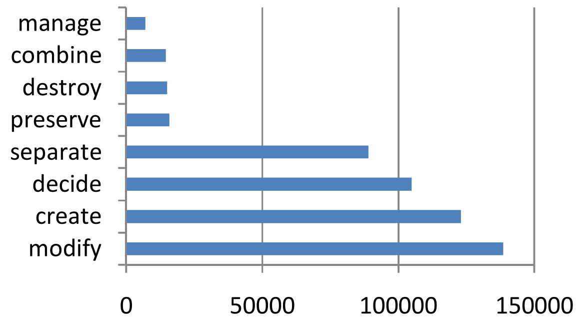

Figure 2. Distribution of Considered Identifiers in the MIT Process Handbook

The empirical analysis has shown that both preconditions for the construction of Web Service-documentation are satisfied. But the idea of using thematic roles is also connected with some difficulties. One is that there is no universally agreed upon set of thematic roles. Many authors have proposed sets of thematic roles for different purposes. Most of these sets differ from each other in their terminology and defined roles they define. Often, the definition of roles intuitively seems to be clear, however in practice their assignment could be difficult or even ambiguous [12].

Because of the different sets of thematic roles in the literature, an appropriate set of roles for software documentation is required. The work of Girardi and Ibrahim has been identified to be the only existing one, addressing this question [13]. They investigated how to improve the retrieval of software by using semantic structures. Therefore they determined a set of thematic roles used in the short descriptions of UNIX-commands. This set of roles has common features and differences with the roles of other authors. Common features among others, are the roles *agent* (entity which performs the action), *location* (place where the action is performed) and *beneficiary* (the entity benefiting from the action). A difference appears if, for example the roles *patient* and *theme* of VerbNet's as well as Saeed's thematic roles are considered. Both roles refer to the affected entity, but the entity is declared as *theme* when the entity is moved by the action and as *patient* when the action changes the entity's state. The roles of Girardi and Ibrahim unify them as *object* which is affected by the action. Girardi and Ibrahim also identified roles which are very software-specific and therefore not explicitly mentioned by other authors, like *comparison* and *manner*. A *comparison* is an entity compared with an object (important for e.g. retrieval-, filtering- and sorting-operations) and the *manner* refers to the mode in which an action is performed (e.g. the silent- or verbose-mode in context of logging or the level of strictness of interpreters).

It has also to be mentioned, that the described approach contains a gap, namely its inability to derive the proper cardinality of parameters and resources. Using thematic roles it can only be derived e.g. that at least one source has to be documented. But there might also be more than one, e.g. several databases to search for a customer. Therefore the developer just gets a set of parameter and resource categories to document, but has to determine their cardinality on its own. A last point to discuss is the impact of the documentation based on thematic roles on the further development of the documented Web Service. Since the documentation is part of the contract between a Web Service's provider and consumer, in many cases the provider has to ensure backward compatibility of the Web Service (to older versions) and therefore to maintain documented, but not required resources. This leads to higher development costs. But one could argue this economical problem does not result from the use of thematic roles, because different stakeholders of documentation could receive different, filtered versions of it. The source of the example from above could be documented as "Customer Relationship Management System deployed on machine with name X" for the internal developers and be filtered as "Customer Relationship Management System" for external developers.

6. Conclusions and outlook

This paper describes a verb-focused approach to identifying the necessary resources of a Web Service Operation which are not explicitly listed in the operation's signature. By using the thematic grid of the verb included in the identifier of an operation and documenting all thematic roles listed in this grid the completeness and therefore the quality of the resulting documentation can be increased. It has been verified empirically that English verbs are commonly used in identifiers of Web Service Operations. Furthermore it has been shown that the Extended Levin Classes cover more of the operation-identifiers and the verbs used in them than the MIT Process Handbook. Thus, the Extended Levin Classes are assumed to be a more appropriate classification framework for Web Service Operations.

The thematic roles identified by Girardi and Ibrahim will be taken as a basis for further investigations. As a first step they will need to be mapped to the Extended Levin Classes. Based on that, an extension of the WSDL-grammar will be developed which directly can support the documentation process by using thematic grids for Web Services. Furthermore, to improve the acceptance of the approach among Web Service developers a software-assistant will be implemented which helps them to document their operations in an intuitive manner.

7. References

[1] P. Clements, F. Bachmann, L.Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, *Documenting Software Architectures - Views and Beyond.*: Addison-Wesley Professional, 2002.

[2] R. Chinnici, J.-J. Moreau and A. Ryman and S. Weerawarana. (2008, June) Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. [Online]. http://www.w3.org/TR/wsdl20/

[3] E. Christensen, F. Curbera, G. Meredith and S. Weerawarana. (2001, March) Web Services Description Language (WSDL) 1.1. [Online]. http://www.w3.org/TR/wsdl

[4] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. (2004, February) Web Services Architecture - W3C Working Group Note. [Online]. http://www.w3.org/TR/ws-arch/

[5] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.-T. Schmidt, A. Sheth and K. Verma. (2005, November) Web Service Semantics - WSDL-S. [Online]. http://www.w3.org/Submission/WSDL-S/

[6] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M.

Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan and K. Sycara. (2004, November) OWL-S: Semantic Markup for Web Services. [Online]. http://www.w3.org/Submission/OWL-S/

[7] D. Oberle, N. Bhatti, S. Brockmans, M. Niemann and C. Janiesch, "Effektive Handhabung von Service-Informationen im Internet der Dienste," *Wirtschaftsinformatik*, vol. 51, no. 05 / 2009, pp. 429-452, 2009.

[8] N. Chomsky, *Lectures on government and binding (Studies in generative grammar).*: Foris Publications, 1981.

[9] D.R. Dowty, "On the Semantic Content of the Notion of the 'Thematic Role'," in *Properties, Types and Meaning*. Dordrecht, The Netherlands: Kluwer Acadamic Publishers, 1989, pp. 69-129.

[10] C.J. Fillmore, "The Case for Case," in *Universals in Linguistic Theory*. New York, USA: Holt, Rinehart and Winston, 1968, pp. 1-90.

[11] J.S. Gruber, Studies in Lexical Relations, 1965, PhD Thesis at Massachusetts Institute of Technology, Dep. of Modern Languages.

[12] J. I. Saeed, *Semantics*, 3rd ed. Malden, Oxford, Carlton: Blackwell, 2009.

[13] M.R. Girardi and B. Ibrahim, "An Approach to Improve the Effectiveness of Software Retrieval," in *Proc. of the 3rd Irvine Software Symposium (ISS'93)*, Irvine, 1993, pp. 89-100.

[14] J. Mendling and J. Recker, "Towards systematic Usage of Labels and Icons in Business Process Models," in *CAiSE 2008 Workshop Proceedings - 12th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2008)*, Montpellier, France, 2008, pp. 1-13.

[15] T.W. Malone, K. Crowston and G.A. Herman, *Organizing Business Knowledge - the MIT Process Handbook*.: The MIT Press, 2003.

[16] B. Levin, *English Verb Classes and Alternations: A Preliminary Investigation*.: University of Chicago Press, 1993.

[17] C. Fellbaum, *WordNet - An Electronical Lexical Database*.: The MIT Press, 1998.

[18] A. Korhonen and T. Briscoe, "Extended Lexical-Semantic Classification of English Verbs," in *HLT-NAACL 2004: Workshop on Computational Lexical Semantics*, Boston, Massachusetts, USA, 2004, pp. 38-45.

[19] K. Kipper Schuler, VERBNET: A Borad-Coverage, Comprehensive Verb Lexikon, 2005, PhD Thesis at University of Pennsylvania.

[20] K. Toutanova and C. D. Manning, "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, 2000, pp. 63-70.

[21] G. K. Zipf, *Human Behaviour and the Principle of Least-Effort*.: Addison-Wesley, Cambridge MA, 1949.