UDC 004.94

CONTEXT WEB SERVICES MATCHING IN THE DISCOVERY TASK RESOLVING. ONTOLOGICAL APPROACHES

Olha Zakharova[0000-0002-9579-2973]

Institute of Software Systems, National Academy of Sciences of Ukraine, 44 Glushkov Pr., Kyiv, 03680, Ukraine

Matching is integral part of all web services tasks. The effective implementation of the process of matching services with a search request, or with each other, ensures successful resolving of the problems of service discovery, building the best coverage of the search request and a complex composite service that implements the business task. All web services have own characteristics, solve different problems, use different methods, but they all also have a certain set of characteristics: functional and process models, formal description language, communication protocol, a common set of element types, such as identifiers, service messages, parameters and etc. This allows to classify services in a certain way, the analysis of these characteristics permits to define the elements and aspects of the comparison process. Today there is a huge number of services, and their number is constantly growing, and the process of services discovery is very complex and multifaceted. It should take into account the structural, syntactic, but, primarily, the semantic suitability of services, to ensure their comparative analysis according to the maximum possible number of characteristics that are significant for matching. It should be step-by-step process and it has be designed in such a way that each step reduces the set of candidates, leaving services with a greater degree of compliance. This requires algorithms that return quantitative estimates for each step, each type of matching, and algorithms that effectively generalize these estimates to define the final values of the proximity of services and the request. These researches are devoted to problems of using the descriptive logics formalisms for web services matching by their contexts, which, as a rule, contain information about the services purposes, the area of use, business functions, etc. That is, it is information representing the semantics of the service, but in text form, that is not convenient for automated processing. There are many studies that try to solve this problem by applying standard text analysis methods to contextual service descriptions. This study proposes an ontological approach to matching web services by context. It is determined the extension of the previously proposed the top-level service DL ontology. It also involves the use of a special ontology of the general textual service description, a fragment of the taxonomy of which is presented in the paper. This ontology should cover all important semantic aspects of contextual descriptions. It have not only promote to determining the matching a service and a request, but it also have to allow semantically categorizing the available services: by subject area, implemented functions, etc.

Keywords: web service, discovery task, matching process, matching types, matching degree defining methods, semantical matching, contextual matching, ontological approaches, top level service ontology, web services categorization and classification, a taxonomy of general textual description of web services, matching of web services by context.

Introduction

The matching process consists in a pairwise comparison of the objects' characteristics with the purpose of establishing their correspondence. From the perspective of solving the web services problems, the main requirements for the comparison process can be defined as follows:

- the matching should not be based only on the comparison of services' descriptions by keywords, it should be taken into account the semantics of services and requests announcements;
- the process must be automated;
- the matching process must be accurate. If the result is returned with a certain degree's value, then it is required a precise definition of the entity "degree of correspondence";
- the process must be efficient and fast. The algorithm should select a small number of services that meet the request best.

Definitions and a general architecture of the matching process

Web services are formally defined objects with a certain set of characteristics, at least, input and output parameters, the used protocol of communication, the representation languages, the type of functional model, and so on. All these characteristics are common to all services and allow to classify them in a certain way at the top level, despite the detailed description.

This allows to filter a set of services at the initial stage for further execution of their costly complex analysis and matching with the request. An example of such a classification is shown in Fig.1.

To provide a constant, timely process of the matching that guarantees an adequate result, its information model must contain:

- a register or data base (DB) of web-services advertisements, their formal definitions, ontologies of services;
- global ontology of the concepts that used in the general services descriptions;
- general service ontology;
- domain ontologies for business-tasks. In the matching process, it can be used additional information resources (DB of the pairs of terms with distance between them, concept hierarchy used in descriptions of the services that are in register, internal terminology), as, for example, implemented in Larks [1].

General classification of approaches for matching web services

Conditionally, it is possible to identify:

- syntactical matching;
- semantical matching;
- contextual matching.

It is clear that to achieve an effective result, the matching process must be an agreed combination of these approaches.

Syntax matching is the technical process that can lead to incorrect conclusions about the results of the matching: syntactically similar services can implement different tasks, and vice versa, services that solve a certain problem, may have different syntax.

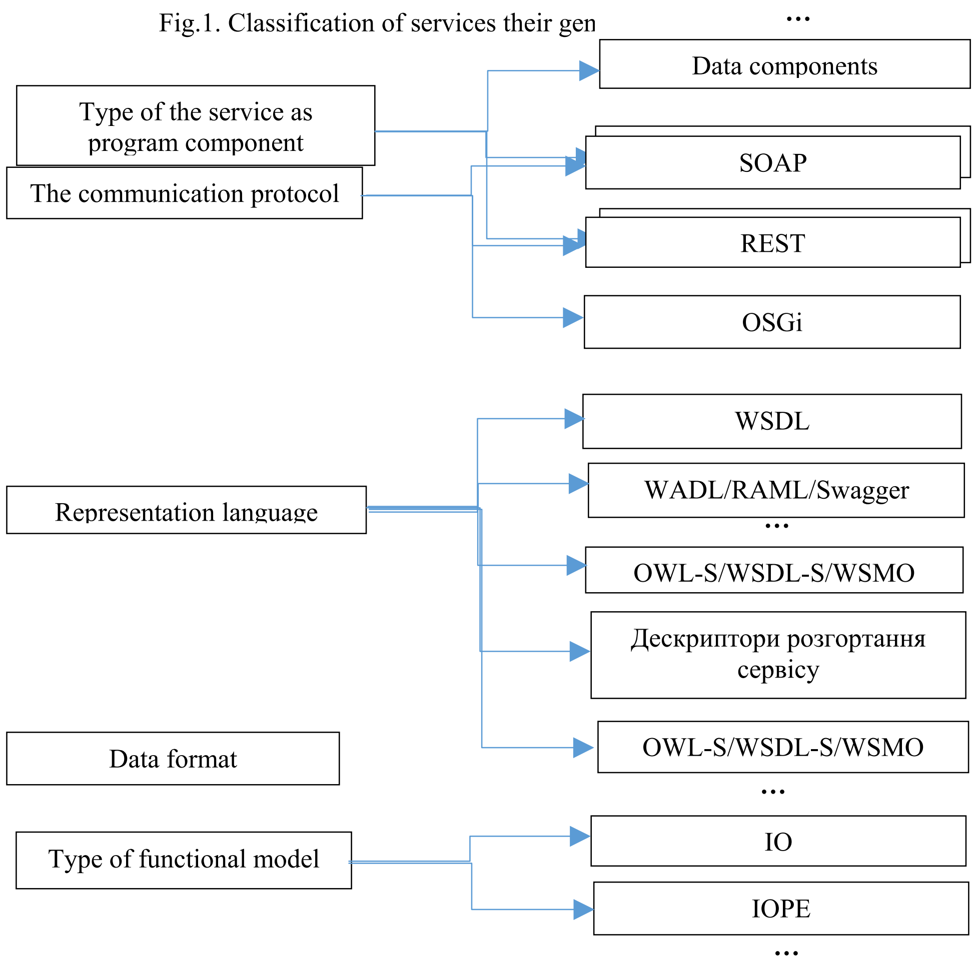
The most interesting is the *semantical matching* of services. Since, primarily, the purpose of the service search is to perform a specific task, and only the semantic description of the service determines its content, goals, functions, etc.

Semantical matching of web-services. Semantical matching approaches may be classified as based on:

- a domain ontology;
- usage of the general ontology;
- syntax and semantic.

Approaches based on a domain ontology consider ontologies as structures of knowledge for determining the attributes of services, their properties and relations between them. The knowledge base (KB) of ontology is used both to determine the request and to define the service. Concepts that were not defined in the original description may be added to it, for example, as semantic annotations, and used in the matching process. To determine them, it is used the knowledge from the domain ontology. It allows getting a more accurate result, namely, to find the semantic correspondence of services and request that do not contain exactly matching attributes, on some abstract level. In addition, such approaches can ensure the categorization of services based on their semantics. Also, they allow to achieve a functional level of categorization based on the ontology, as well as provide an opportunity to improve the semantic description of the request, which, in turn, provides a better result in searching services.

Fig.1. Classification of services their gen



It should be noted that there are many methods of matching based on the domain ontology, which offer a variety of algorithms for determining the proximity of services and request: methods which use different models of services representation, based on calculation of semantic distance, compare different characteristics and so on. Their advantages are the focus on automated services identification, integration and discovery, and the use of semantics, which allow to improve the relevance of their discovery. However, most of these approaches are based on specific service representation languages, such as OWL-S, WSMO or WSDL-S, which require from end users a certain level of knowledge. Also, the service requester may not fully possess the knowledge contained in the domain ontology, may not know all the conditions associated with the request, and, thus, the request may not be correctly defined. This can lead to the loss of many relevant services in the discovery process. Another limitation of existing approaches is that they are based on the assumption that service providers and service requesters use the same domain ontology to describe the service and request, but in real life, as a rule, this is not the case. It is impossible to overcome this difference between ontologies without ontology mapping techniques.

Approaches based on the general ontology allow users adding semantic information to services, without semantic annotating. So, in contrast to the traditional approach, they do not require significant costs for ontology management and semantic annotation and are independent of the domain ontology. It is a question of definition of semantic tags on a part of web services messages. These approaches are based on implicit semantics of web services. To form the result of the matching they use a general data base and semantic resoners. In this case, to semantize a service, it is built a service ontology. It may be a special self-defined ontology or a commonly used ontology. The usage of the general ontology simplifies applying the methods of this group and provides the possibility of their reuse. Such a well-known general-usage dictionary use is, for example, WordNet. Its advantage is that the terms and concepts in it have precise semantics. WordNet can be used as a service ontology, its concepts (semantic words) can be used to annotate a library of services, the service and its interface. Today, there are many examples of using WordNet for semantic annotation of web services [2-5]. This is mainly due to the independence of WordNet from a particular domain, what is its main advantage.

Approaches based on syntax and semantics. This group includes services discovery processes based on the matching functional and non-functional requirements of the services and the request ([17]) and defining the quality of services based on special metrics. These can be purely syntactic approaches, purely semantic approaches, or methods that combine both syntactic and semantic algorithms. At the same time, they can use different ontologies in the matching process: an ontology for integrating quality standards and defining service properties; an ontology for modeling relationship between these standards and users preferences to rank services by functional characteristics; an ontology that allows to establish the relationship between quality models and the web services functionality, etc.

Web services discovery process, in this case, may combine different types of search: by context, by functionality and by quality metrics. The discovery process may take into account a search of synonyms. When establishing the appropriateness, the approaches of this group can use both the specifications of services and their signaturs, combine logical and non-logical matching.

Context approaches. Business tasks strongly depend on the situational context in which they are performed, but the technologies of semantic web services usually do not encourage to present such situations in the domain. Moreover, the description of the complex specific situation, with all its aspects, is a very valuable task and may never reach a sufficient semantical expressiveness. No situations in the real world can completely match, but to ensure adaptability to the context the finite sets of semantically defined descriptions of parameters must correspond each other. This group of methods guarantees the most accurate automated discovery based on a comprehensive semantic description of the request and service and taking into account the situational context or user context [7].

Understanding the context is a key component in any common system and provides system intelligence by allowing computing devices to receive appropriate and timely decisions on behalf of the user. Such systems must adapt to changes and variations in the user context [8]. But the notion of context is too wide, as the context can be any information related to the user or software system [9]. It is very difficult to model, especially the only way for different applications. Using it in automated algorithms is also not trivial. This causes the complexity of mass-use of context-sensitive approaches and a lot of problems associated with their usage to solve problems of web services.

Types of matchings and a classification of matching methods

The most of the web service tasks are based on defining effective approaches to discovery needed services according to features of their representation. The set and features of the items that is used to determine the services and the request, the model of service and request and the specifics of the business task determine the different types of matching:

- *Matching natural language descriptions of services and querry (contexts).* WSDL syntax allows creating text descriptions of services, their types and operations that groupped in tegs <Documentation>. So, in this case, the problem is to determine a proximity degree of text descriptions of web services. It is a traditional task of an information search.
- *Structural matching services and request (matching profiles).* Structural matching of web services consist of comparison of their functional models. There are different types of web service functional models. The most widespread is IOPE model. In IOPE model, functional features of services are defined by inputs/outputs, preconditions and effects. Namely, IO (inputs, outputs) provide syntactical information of web services, as PE (preconditions and effects) present their semantics.

The methods used to match inputs/outputs and those used for preconditions and effects are different. Semantics presented by inputs/outputs and preconditions/effects are different too. To date, significant results have been achieved in matching web services by inputs/outputs but no effective approaches to match them by preconditions and effects exist. So, consider PE matching as particular category.

- *Matching services and a request by preconditions and effects (matching conditions and limitations)*
- *Matching operations of services.* It is a sequential step in solving the general problem of service discovery. The degree of relevance of operations is defined based on the degree to which their incoming and outgoing messages are similar in the previous steps of the matching. The general value of the degree of proximity of two services is defined by establishing the pairwise correspondence of their operations, which maximizes the total amount of matches for individual pairs.
- *Semantical structure matching by operation identifiers (matching signatures).* The names of types, operations and services, as usual, present semantics of the service features. This type of matching is the process of defining semantical similarity of data type identifiers, operations and services [10].

According to the listed types of matching, the following groups of methods can be distinguished.

Structural matching methods. Today, most web services have a profile representation in WSDL [11]. So, most structural matching approaches are based on the matching input and output WSDL specifications. The last includes a matching sets of operations offered by services, which, in turn, is based on comparison of the structures of input and output messages of operations, based on comparing the data types of objects that are transmitted in these messages. The result of this comparison is the value of the proximity degrees of all data types of the source and the target. In [12] authors proposed an approach where these values form a matrix, which can be used to define the overall estimate of matches of services both on input and output parameters, and as a whole. It should be noted that, first, the data types of web services in WSDL are XML items and they can have quite complex structures. Second, WSDL service specifications do not include their semantics, and this mapping is purely technical.

Other subgroup of structural matching approaches is methods based on DAML-S ontologies. DAML-S [13] is built based on logics and supports semantical information specification in RDF format. DAML-S provides service representation that cover behavioral specifications of operations. DAML-S profile of the service is defined by IOPE model. DAML-S enables discovering services through matching specifications. Different approaches are known to implement algorithms of semantical matching based on DAML-S ontologies [14]. What they have in common is the announcements of the services and the request refer to DAML concepts and related semantics. DAML enables logical inference on the hierarchies of categories in the matching. It leads to the recognition of semantic similarity, regardless of syntactic differences and distinctions in the modeling abstractions in the services announcements. Also, DAML provides the accuracy of the matching: similarity is as it may be inherited from DAML ontologies used in the registry. Additionally, semantics of the DAML-S descriptions allow defining rank functions for proximity degrees. At least, DAML-S allows both to compare two services and to formally prove that found and target services don't conflict. The general architecture of the matching based on DAML-S may be similar to shown in fig.2.

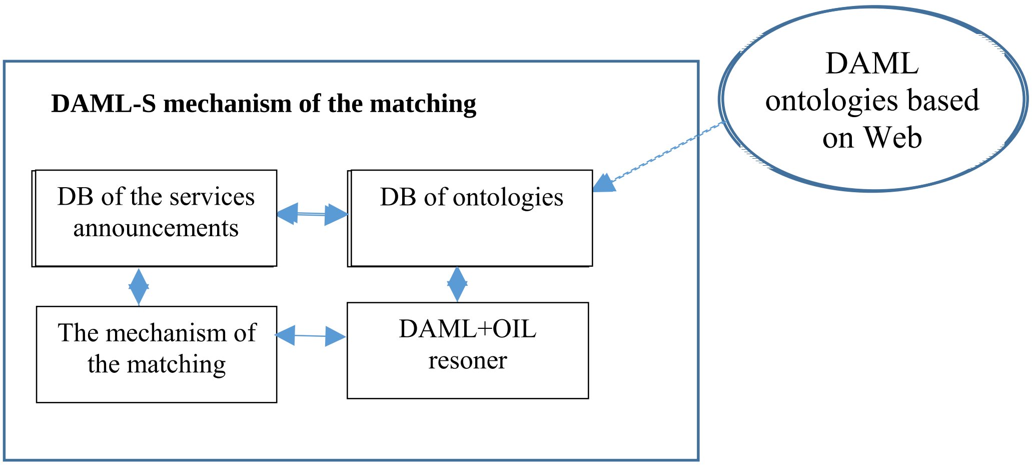


Fig.2. The architecture of the DAML-S matching

Similar architecture may be used also in another ontological approaches. In the [15] it is presented the PE matching algorithm based on DL SHOIN+(D). Its core is next: if two services are presented by PE model based on the same acyclic Tbox: $S_r=(pre_r,effect_r)$ та $S_a=(pre_a,effect_a)$, where S_r service-request, and S_a – the real service, then S_a is matching S_r, if:

- Preconditions of the announced service are simpler as preconditions of the request, or, at least, these conditions may be easily satisfied.
- The effects of the service aren't less than request's effects.

Formally, we can formulate the task of structural matching for the service and the request that are presented by IOPE model as follow. Let the service is *S={s| s = (CI; I; O; A)}* from repository □□, where *CI* – its preconditions, *I* and *O* – finite sets of inputs and outputs, respectively, *A* – its effects, and the request *Q* is presented as the searched service *Q= (CI'; I'; A'; O')*, where *CI'* - preconditions, *I', O'* – the finite sets of inputs and outputs, respectively, *A'* – its effects, then the service *S* corresponds to *Q,* if [16] *I□I', O'□O,* and preconditions of the request *CI'* is wider then the preconditions of the service *CI*, so the fulfillment of the request conditions attracts the fulfillment of the service conditions, and for effects, vise versa, the execution of the effects of the service attracts the execution of the effects of the request.

The methods for matching text descriptions of the services. The main focus of previous researches is resolving the discovery and composition tasks for services presented with DL ontologies [17, 18, 19, 20]. But proposed approaches, at all, didn't take into account a service context. But it is a very important aspect when matching the service and the target. Usual WSDL profile of the web service include the teg *Documentation* that allows to define the service context with a valuable semantical information. Its analysis enables to prevent including of services that solve completely different business problems into the set of candidates for the composition.

At the same time, to use contextual descriptions of the web services in resolving their problems we need effective automated methods for analysizing and matching textual information. The most widespread methods for solving problems of determining the text descriptions similarity are methods based on vector space models. In other words, it is carried out the search for the services most similar to the input description of the corresponding vectors. The corresponding vectors of the document and of the request are matched and the similarity values are calculated. To obtain more adequate estimates, weights are often used. Also, various lexical dictionaries are used (for example, dictionaries of synonyms) in the searching and matching processes. As well, they are known the extensions of methods of vector analysis [21] that are based on the use of lexical database WordNet [22 - 26] as the basis of vector space. The highest overall score indicates a greater degree of similarity between the original and target specifications. This type of matching is similar to the signatures matching in the searching software components [27] and the matching with the document template in the information search.

The disadvantages of such methods for determining the similarity of textual information [28] are, primarily, their complexity, and, secondly, they actually determine only the lexical similarity of the text to the target document, but the same content can be expressed in natural language different ways.

Recently, they have been actively appearing the ontological methods of data semantization, including textual ones. One of such approaches is to use taxonomies of concepts to calculate semantic distances between concepts. But taxonomy defines only the generalization / detailing relationships based on the concept definitions through roles and sets of attributes. This may not be enough, for a deeper understanding of semantics it is advisable to identify additional associative relationships between concepts. In [4], it is used the so-called weighted associative network - a semantic network with concepts in nodes and directed weighted edges between them. In this semantic network, three types of binary weighted relations are allowed: (1) generalization, (2) detailization, and (3) positive association among the concepts [29] - the most common relationship between the concepts. Such associative relations can be found, for example, based on WordNet ontology.

Establishing the semantical similarity of identifiers of data types, operations and services. This process, as a rule, is reduced to establishing the similarity of identifiers. It is the similar to process of structural matching of WSDL descriptions. The result may be, for example, the matrix of assessments of similarities [30] for all pair combinations of data types of the source and target services. The result of a further comparison of operations may be to determine the degree of correspondence of the two operations based on the semantic distance between the names of operations, and sets of their parameters in terms of identifiers that they include. The general indicator of similarity is calculated by matching the names of services and pairwise similarity of their operations, which maximizes the total amount of coincidences of the pairs

Elements of the matching process which may be defined by DL formalisms. Based on certain types of matching and types of semantics [9], we can conclude that the matching process should cover next elements of services:

1) a general contextual description of the service;
2) inputs of the service;
3) outputs of the service;
4) pre-conditions of the service;
5) post-conditions of the service;
6) effects of the service;
7) objects produced by effects of services;
8) variables of internal operations of the services;
9) the results of service operations;
10) conditions for performing service operations;
11) messages of the service.

DL may be used both to define semantical description of the service and the target and to obtain results of matching. Any formalization of service semantics with DL allows to use its reasoners at different stages of matching. So, matching the services by its inputs and outputs is a standart problem of DL inference.

Using DL tools to resolve the problem of service matching by their contexts

Matching general contextual descriptions should be one of the initial stages in solving the discovery problem. Its purpose is to cut off services that semantically don't match the request: have other goals, work in another applied field with other data, etc. A taxonomy for resolving such problem, in some aspects, have to be similar to DB used for matching texts. But, taking into account possibility of formal representation of the description, it is desirable that it not only helps to determine the proximity of the target and available descriptions, but, primarily, also allows to categorize the available services: by domain, by types of functions that are implemented, etc., so, covers all important semantic aspects of contextual descriptions. The possible structure of the taxonomy is shown in fig.4.

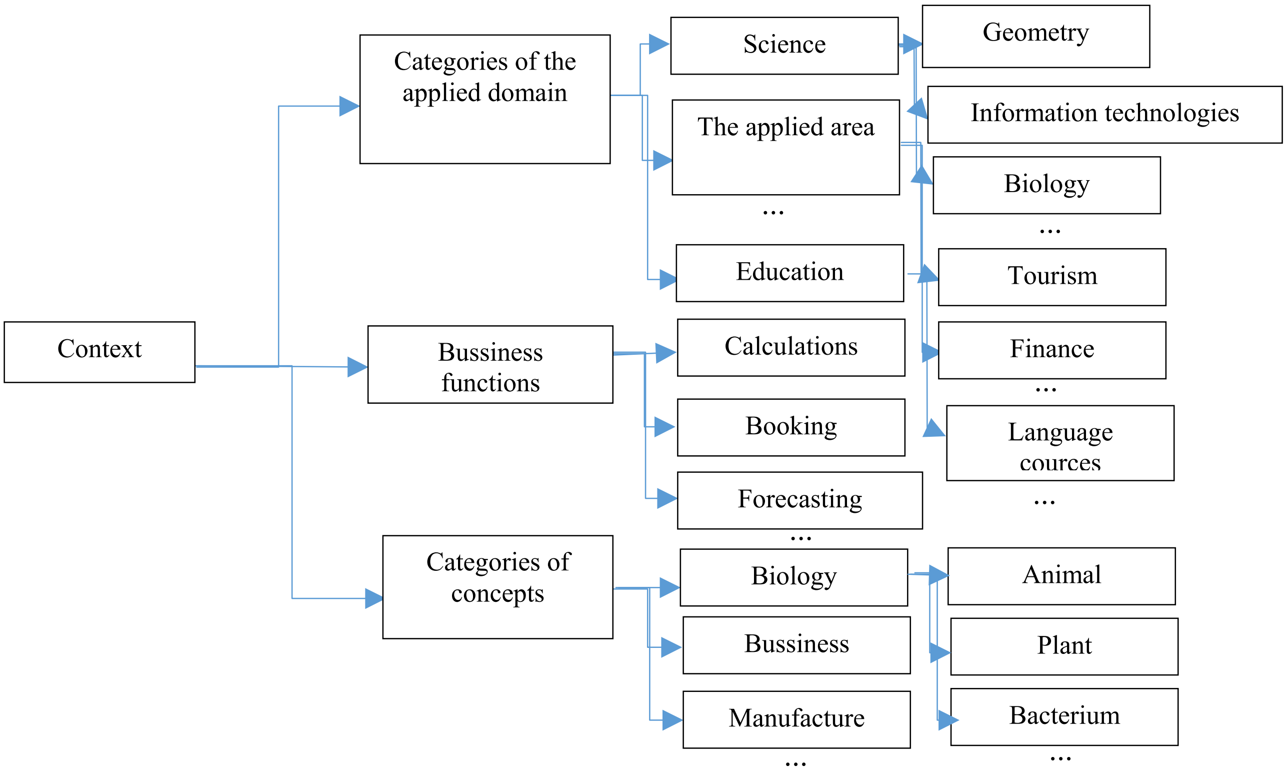


Fig.4. The architecture of the taxonomy of the general textual service description

Based on the taxonomy of the concepts that characterize the service content, it makes sense to define associative relations, which, for example, will allow to determine possible relations between business functions. It is very important to define associative lexical relations based on concept categories, primarily, synonimy. The ontology corresponding to shown fragment of the architecture allows to define general aspects of service and request descriptions. The fragment of the DL ontology ***DocumentationOntology*** shown below:

TBox

ServiceContext
BusinessDomenCategory □ ServiceContext
...
BFunctionType □ ServiceContext
MainEntitiesTypes □ ServiceContext
ClassicScience □ BusinessDomenCategory
AppliedSphere □ BusinessDomenCategory
Geometry □ ClassicScience
IT □ ClassicScience
Algebra □ ClassicScience
…
Turism □ AppliedSphere
Trade □ AppliedSphere
Finance □ AppliedSphere
…
LanguageCources □ Education
…
Calculation □ BFunctionType
…
Mathematic □ MainEntitiesTypes
Zoology □ MainEntitiesTypes
…
Average □ Mathematic
Square □ Mathematic
…

Shown fragment of the taxonomy is only a demonstration example of a general structure of concepts for describing the context. Main purpose of such ontology is to define maximally full general system of concepts that enable a structured semantical description of the context of any service, regardless of the scope of its application or business purpose. But only as both the system of concepts and the appropriate knowledge base are adequat and have a high degree of completeness, so such ontology can assist to the effective solving of web service problems. Therefore, its development is a separate, not trivial task that requires the involvement of experts.

DL description of the web service should be defined based on both the service ontology of the top-level and on applied ontologies, which determine the semantics of inputs, outputs, preconditions, effects, etc. In [17] it has already been proposed the top-level service ontology for the IOPE model - ***ServiceOntology***:

TBox

Service, InputParameter, OutputParameter, PreCondition, PostCondition, Parameter,
Name, Value, Type
Service □ has.InputParameter;
Service □ has.OutputParameter
Service □ has.PreCondition;
PreCondition □ Condition
PostCondition □ Condition;
Condition □ hasValue.Boolean
Service □ has.PostCondition;
InputParameter □ Parameter
OutputParameter □ Parameter;
Parameter □ has.Name
Parameter □ has.Value;
Type = {String, Numeric, Boolean}
I_S □ *InputParameter*
O_S □ *OutputParameter;*
CI_S □ *PreCondition;*
CO_S □ *PostCondition;*
Name □ hasType.{String}; Value □ hasType.Type

The proposed ontology actually defines the basic concepts of a service. Ontology classes allow to determine the main items of the IOPE model. However, this ontology does not provide the possibility of formalization of the general contextual description of the service. To do this, it is necessary not only to expand the ontology of top-level service, but also to define the links with applied ontologies and general ontology of concepts. It would allow to categorize the service in a certain way and to carry out the matching of services not only by inputs and outputs, but also by contexts, which, as a rule, provide semantic information. Corresponding extension of TBox ontology ***ServiceOntology*** is presented below:

Documentation, DocItem
Service □ has.DocItem;
Service □ has.Documentation
DocItem □ Documentation;
DocItem □ has.ItemName;
DocItem □ has.ItemValue;
*DocItem □ BusinessDomenCategory / *link to* ***DocumentationOntology***
DocItem □ BFunctionType / link to* ***DocumentationOntology***
*DocItem □ MainEntitiesTypes / *link to* ***DocumentationOntology***
ItemName □ hasType.{String};
ItemValue □ hasType.Type

Definition of the task of matching by context. Given: S – a service and Q – a request. Let's define the matching task as:

S □ Service, Q □ Service
S □ has.DocItem
Q □ has.DocItem
DocItemS □ DocItem, DocItemQ □ DocItem

The service *S meets* the request Q by context if its definition includes descriptions of all semantics that defined in the request's context, namely: *DocItemQ □ DocItemS.*

To solve the problem using DL, each service, as well as the request, is considered as a specific ontology with relations to many other ontologies. And the decision on the inclusion of concepts is made by the reasoner based on the analysis of the whole set of related concepts and different types of relations between them (including associative relations).

Same as in the case of matching by inputs/outputs, we can establish different degrees of matching by the context. As general indicators of correspondence we can define:

- ***Exact***. Exact match is if the concepts are equivalent *DocItemQ ≡ DocItemS.*
- ***Plug In.*** *DocItemQ □ DocItemS* – condition which is sufficient to make a decision about the relevance of the request and the service: the service in its context contains all the concepts requested.
- ***Subsumes.*** *DocItemS □ DocItemQ*, in this case the service according to the context is not completely match the request, because its description is not enough complete.
- ***Fail*** - failure, it is not found any *subsumption* between *DocItemS* and *DocItemQ*.
- ***Intersection***. *DocItemIntersection ≡ DocItemQ □ DocItemS.* Considering the specifics of the individuals *DocItem* and the concept description, all services the contextual description of which has a non-empty intersection with the description in the request should be taken into account. And percentage estimations of this intersection, for example, the relation of cardinality of the set *DocItemIntersection* and cardinality of the set *DocItemQ* may more exact define the degree of matching. It is probable this estimate should be more complex and should take into account the categories of items included to intersection, or calculate the proximity of the *DocItemS* and *DocItemQ* concepts*.* This is a separate difficult task and requires additional research and the development of special approaches to create such estimates, to estimate their complexity and feasibility of application.

Conclusions

Taking into account the huge number of existing services and the dynamics of their creation, it is clear that the task of finding the necessary service or services for resolving the business problem is difficult and requires effective approaches to discovery services-candidates and to establish the degree of their compliance, filtration, etc. The matching services should be step-by-step process, and at each step the set of candidates is significantly narrowed. Matching by context is a necessary step in the overall process and allows cutting off services that may have a similar set of parameters, but implement different functions.

Advisability of using DL is justified at all stages of solving the discovery problem. It is the tool that provides the formal definition of the service and reasoners which enable the logical inherits. Each service is represented by DL ontology based on the top-level ontology. The ontology of the service is based on the knowledge base of integrated DL ontology of the domain. Ontologies by themselves do not describe the possibilities of services, but they provide additional concepts that allow to correctly understand the specification of the service possibilities. Existing DL reasoners allow to carry out automatic matching. These are the standart problems of DL inherit. In this work it is proposed the approach for matching web services by contextual descriptions based on DL. Creation and usage of the special ontology of the most common concept types enable increasing the efficiency of discovery solutions and permit to categorize /classify the services: by usage area, realised functions, etc. The use of such ontology in the annotation or formal definition of services can significantly simplify the task of identifying and matching services. And when using the approaches of formal definition and matching web services based on DL ontologies, it enables automatic usage of DL representation of such ontology for resolving problems with DL reasoners.

DL is a quite complex tool to formalize the service descriptions by their developers. But, taking into account that descriptions of semantical web services profiles are WSDL documents, which are supplemented by semantic annotations (so they are an ordinary XMLs), it is possible and efficient the automated parsing of this XML-definition and constracting the ontology of the concrete service based on top-level service ontology.

References

1. K. Sycara, S. Widoff, M. Klusch and J. Lu. "LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace". Autonomous Agents and Multi-Agent Systems, 5, 173–203, 2002.
2. R. Karimpour, F. Taghiyareh, "Conceptual discovery of web services using WordNet," in Service Computing conference, Singapore, Dec. 2009.
3. G. Ganapathy, C. Surianarayanan, "An approach to identify candidate services for semantic web service discovery," in IEEE international conference on service-oriented computing and applications, Perth,WA, Dec. 2010.
4. Y. Peng, "Two levels semantic web service discovery," in seventh international conference on fuzzy systems and knowledge discovery, Yantai, Shandong, Aug. 2010.
5. Joerq Becker, Oliver Mueller, Manuel Woditsch, "An ontology-based natural language service discovery engine-design and experimental evaluation," in the proceeding of the European conference on information systems, Pretoria, South Africa, 2010.
6. N.Levy, F. Losavio, A. Matteo, A. Ramdane-Cherif, H. Hadj Salem, "Quality standards for ontology web service discovery," in international conference with peer:XXXVII CLEI, Quito, Ecuador, October 2011.
7. Hua Xiao, Ying Zou, L. Ng, L. Nigul, "An approach for context-aware service discovery and recommendation," in IEEE international conference on web services, Miami, FL, July 2010.
8. Peyman TalebiFard, Victor C.M. Leung, "A data fusion approach to context-aware service delivery in heterogeneous," in 8th international conference on mobile web information systems, 2011.
9. Wenge Rong, Kecheng Liu, "A survey of context aware web service discovery: from user's perspective," in SOSE '10 proceedings of the 2010 fifth IEEE international symposium on service oriented system engineering, Washington, DC, USA, 2010.
10. A. M. Zaremski and J. M. Wing. "Signature Matching: a Tool for Using Software Libraries". ACM Transactions on Software Engineering and Methodology, Vol. 4 No. 2, 146-170, Apr. 1995.
11. Web Services Description Language (WSDL) (WSDL) http://www.w3.org/TR/wsdl
12. http://dit.unitn.it/~p2p/RelatedWork/Matching/icsoc2004.pdf
13. The DARPA Agent Markup Language Homepage. http://www.daml.org/
14. https://www.researchgate.net/publication/221466239_ Semantic_Matching_ of_Web_ Services_Capabilities
15. Hai Wang, Zengzhi Li. A Semantic Matchmaking Method of Web Services Based On SHOIN+(D). /Institute of Computer System Structure and Networks School of Electronics & Information Engineering, Xi'an Jiaotong University, Xi'an Shaanxi 710049, PR China hwang@mailst.xjtu.edu.cn, lzz@mail.xjtu.edu.cn.
16. Srividya Kona, Ajay Bansal, Gopal Gupta Department of Computer Science The University of Texas at Dallas Richardson, TX 75083, Thomas D. Hite Metallect Corp. 2400 Dallas Parkway Plano, TX 75093Automatic Composition of Semantic Web Services.
17. Zakharova O. Defining and resolving Web-services discovery problems using description logics formalism. Problems in programming. 2017. N 4. P. 66–78.
18. https://www.researchgate.net/publication/221466239_Semantic_Matching_of_Web_ Services_Capabilities
19. http://dspace.nbuv.gov.ua/bitstream/handle/123456789/144567/07-Zakharova.pdf? sequence=1
20. Zakharova O. Semantic annotation of web services. Problems in programming. 2015. N 4. P. 49–57.
21. G. Salton, A. Wong and C.S. Yang. "A vector-space model for information retrieval", In Journal of the American Society for Information Science, Vol. 18. November 1975, 13-620. ACM Press.
22. E. Voorhees. "Using WordNet for Text Retrieval", in C.Fellbaum (ed.), WordNet: An Electronic Lexical Database 1998, The MIT Press, Cambridge, MA. 1999, 285-303.
23. G.A. Miller, R. Beckwith, C. Felbaum, D. Gross and K. Miller, "Introduction to WordNet: An On-line Lexical Database", International Journal of Lexicography, Vol. 3, No.4, 1990, 235-244.
24. G.A. Miller, "Nouns in WordNet: A Lexical Inheritance System", International Journal of Lexicography, Vol3, No.4, 1990, 245-264.
25. R. Mandala, T. Takenobu and T. Hozumi. "The Use of WordNet in Information Retrieval," in Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, 1998, 31-37.
26. R. Richardson and A.F. Smeaton. "Using WordNet in a knowledge-based approach to information retrieval." Dublin City University School of Computer Applications Working Paper CA-0395.
27. J. Purtilo and J.M. Atlee. "Module Reuse by Interface Adaptation". Software Practice and Experience, Vol. 21, No. 6, 1991, 539-556.
28. C. Faloutsos, and D.W. Oard. "A survey of Information Retrieval and Filtering Methods, University of Maryland". Technical Report CS-TR-3514, August 1995.
29. P. Fankhauser, M. Kracker, and E. J. Neuhold, "Semantic vs. structural resemblance of classes. Special issue: Semantic issues in multidatabase systems," ACM SIGMOD RECORD, vol. 20, no. 4, pp. 59–63, 1991.
30. Y. Wang and E. Stroulia. "Flexible Interface Matching for Web-Service Discovery". In Proceedings of 4th International Conference on Web Information Systems Engineering, December!10th - 12th, 2003 (to appear).

About author:
Olha Zakharova,
PhD,
Researcher.
Scientific Ukrainian publications – 30.
http://orcid.org/0000-0002-9579-2973.

The author's place of work:
Institute of Software Systems of NAS of Ukraine,
Glushkova ave, 40.
Tel.: +38(044)526 5139.
E-mail: ozakharova68@gmail.com.
Mob.tel.: +38(068)5947560