SOF: A Slight Ontology Framework Based on Meta-modeling for Change Management

Li Na Fang, Sheng Qun Tang, Ru Liang Xiao, Ling Li, You Wei Xu,
Yang Xu, Xin Guo Deng, and Wei Qing Chen

State Key Lab of Software Engineering, Wuhan University, 430072, Wuhan, China
dpetfln@sina.com

Abstract. The importance of efficient E-Government services change management system is increasingly due to the evolution of it. But, the most system management tasks are still performed manually. This can be easier error-made, high time-consuming and more human-needed. So we present a Slight-Ontology-Framework (SOF) to perform the semi-automatic change management. The main ideas can be shown as the following aspects: One, it uses a set of ontologies to describe E-Government services and introduces meta-modeling theory to analyze the features of changes. Two, according to these services' characters, it reduces the description capability of OWL-S and combined itself with Business Process Modeling theory to make it higher pliability and easier system-implemented. Even though we use E-Government domain as the example, the approach is a general solution in other domains.

Keywords: Meta-ontology, semantic web services, framework.

1 Introduction

The most important challenge for the E-Government services is adapted themselves to the complex changes in their environment, besides in their internal structures and processes. The new ways of working with citizens, enterprises, or other administrations efficiently should be concerned by E-Government services. They need the continuing change management to perform the dynamic modification problem in time. Note that, such dynamical refinement changes are not to rewrite project. It is a high level modification which must meet the syntactic and semantic consistency of changed services. Here, we present an approach to resolve it.

This approach uses the semantic technology to enrich the current implementing mechanisms of E-Government service process to support a more efficient management. In the most current Web Services description and Web Services composition languages, the semantic expressivity on the level of business processes, is lacked [1]. Therefore, specifications of domain specific constraints need to be presented, which must be taken into account during the construction of it, and business process flow must be defined at the abstract task levels without consideration of the details of specific service bindings and execution flow. To model this abstract representation of services, a set of ontologies is used to describe them, besides the

consistency of service descriptions, possible changes, as well as their resolutions. Therefore, the evolution of distributed and independent ontologies [2] is the study focus in our approach. Even though the approach is presented in the application domain of E-Government services, it is a general solution in other domains that uses semantic web services.

The paper is organized as follows: In section 2, SOF is introduced. Then, the approach is presented in section 3. The partial implement is given in section 4. Before the conclusion (section 6), we present an overview of related work (section 5).

2 SOF: A Slight Ontology Framework Based on Meta-modeling

In this section, a new ontology framework is given. A set of ontologies is introduced to describe services. Meta-ontology is used to model the dependency between business rules and service implementation in order to define the business process flow about static services and the propagation of changes.

2.1 Ontologies Used to Model Services

SOF uses the evolution of meta-ontology to manage change. So, a set of ontologies, which are used to model services, is presented. Moreover, we classify them as follows: Meta-ontology which contains the entities used to describe service, Domain ontology which contains the special domain knowledge, Service ontology to describe the concrete services, Organization ontology which contains structure concepts of government organization and Law ontology to model the structure of the law document. The relationships between them are shown in Fig. 1.

Fig. 1. The top level ontologies used to describe services

The aim of this management system is to assist public administrators to get rid of a great amount of management tasks to improve maintenance efficiency and reduce cost. Here, we do not focus on the dynamic service composition, but the static web services and the precise definition of business rules (i.e. law) in addition to the dynamic bindings of services during the execution of static services. So in the following part of this section, we introduce the two major parts of meta-ontology: Profile part which is used for service discovery and the process part which is used for describing the business process flow.

“Reference” is added, as well as properties related with it. The property “hasFirst” and “ConsistOf” are presented to describe the composite service flow. The control constructs is defined as follows: sequence, split, join and if-then according to the specific of E-Government domain. Moreover, some properties between services and control constructs are given. The details can be shown in Fig. 2.

3 Approach

The change management is argued in this section as follows: Firstly, changes and consistency are defined. Then the propagation of changes is introduced from business rules to services and within services.

3.1 Changes and Consistency

To manage change efficiently, the meta-modeling is used to analyze the features of ontologies. Changes are classified into two kinds: the basic changes (Add and Sub) and the complex changes(i.e. composed change) such as modify, which can be achieved by the composition of basic changes. For example, a “ModifyConcept” change can be achieved by the sequence composition of previous “SubConcept” change and next “AddConcept” change.

But we need to extend the granularity level of changes to make a better management of changes in a service description. Changes should be defined about concepts of service and relations among them, besides the concept of input, output, preconditions etc. of a service. A full set of basic change can be defined in the following table 1.

Table 1. The set of basic change of ontologies used by SOF

	Additive changes	Subtractive changes
Service	AddService	SubService
ServiceSpecialization	AddServiceSpecialization	SubServiceSpecialization
Input	AddInput	SubInput
Output	AddOutput	SubOutput
Precondition	AddPrecondition	SubPrecondition
Result	AddResult	SubResult
Business Rule	AddBusinessRule	SubBusinessRule
Human	AddHuman	SubHuman
Equipment	AddEquipment	SubEquipment
Software	AddSoftware	SubSoftware
PreviousConnection	AddPreviousConnection	SubPreviousConnection
NextConnection	AddNextConnection	SubNextConnection

To define the consistency of ontologies in SOF, the consistency of ontology should be defined. The reason is that ontologies are a kind of Ontology used to model services. So, Ontology Consistency [5] is defined as the following way: An Ontology is consistent with the respect to its model if and only if it preserves the constraints defined for the underlying ontology model.

Moreover, ontologies may include other ontologies. The dependent ontology consistency is defined as following [6]: A dependent ontology is consistent if the ontology itself and all its included ontologies, observed alone and independently of the ontologies in which they are reused, are ontology consistent.

Finally, we give the consistency of semantic web services as the following way: A semantic web service is a consistent service only if its description is dependent ontology consistent and the additional constraints (C1-C11) are satisfied. The constraints are defined as follows:

- C1. Each service has to have a reference to at least one business rule.
- C2. Each service has to have at least one human or equipment that controls its execution.
- C3. Each service has to have at least one software component attached to it.
- C4. Each service has to have at least one input and one output.
- C5. Each service input has to be either output of some other service or is specified by the end-user.
- C6. If the input of a service is the output of another service, then it has to be subsumed by this output.
- C7. If the input of a service subsumes the input of the next service, then its preconditions have to subsume the preconditions of the next one.
- C8. If two services are subsumed by the same service, then their preconditions have to be disjoint.
- C9. If a service specialize another service, one of its parameters (i.e. inputs, outputs, preconditions or results) has to be different. The difference can be achieved either through the subsumption relation with the corresponding counterpart or by introducing a new one.
- C10. Inputs, outputs, preconditions and postconditions have to be from the domain ontology.
- C11. Any specialization of the service S1 must always be a predecessor of any specialization of the service S2, where S1 and S2 are two services defined in the Meta Ontology and their order is given in advance (i.e. S1 precedes S2).

Semantic web services must obey the constraints defined in this set of consistency constraints. Note that, all constraints should be defined formally to validate the description of services automatically. Services, which are defined in the set of consistency constraints of services, must obey the constraints.

3.2 Changes Management Process

The aim of change management process is to find the weak places of descriptions of services, by considering changes in the business rules and the impact on consistency, and discover inconsistencies in service descriptions, which can be repaired to improve the agreement of ontology with the business rules. So, a propagation procedure is defined to make managers know changed business rules which may

spring some inconsistencies. To implement this propagation, changes should be formalized and automatically performed.

The capability, which the system must have, is that it should automatically identify problems in services or their descriptions, and rank them in sequence according to the importance. Once a problem arise, the system should assist them to identify resources of it, analyze and give solution and, if possible, to help them decide which solution should be chosen to resolve it.

A change management process is defined, which is shown in Fig. 3, as follows: Firstly, it represents a request for a change formally and explicitly. Secondly, to prevent inconsistencies raised by additional changes, changes are preserved to guarantee the consistency of services. Thirdly, the requested changes and additional changes are performed during the change implementation step. Finally, all dependent knowledge items are found and updated in the change propagation phase.

Fig. 3. Four Steps of the change management process

4 The Partial Implementation of SOF

SOF is a slight ontology framework to manage changes. The system based on SOF has been developed which is much more than a standard framework for creating, modifying, querying and storing ontology-based description of semantic web services. It supports the change evaluation management which includes modeling services, discovering services, compositing services as well as reconfiguring services. The SOF platform is roughly presented in Fig. 4. It can be divided into three lays:

1. **Services Application Lay:** It realizes UI applications and provides interfaces to Service Editor used to edit the semantic description of E-Government services, Visual Ontology Editor used to edit the domain ontology, organization ontology and law ontology, and Service Registry.
2. **Middleware API Lay:** It is the core part of the change evaluation management architecture. In Fig. 4, it shows the main including modules: (i) service management module; (ii) consistency check module; (iii) change manage module, which is the core part of middleware API lay, including change implementation submodule, change maintenance submodule, change propagation submodule and evaluation submodule; (iv) service implement module.
3. **Data Storage Lay:** It provides data storage facilities which include storages of service data, ontology data, reason data change data and log data.

Fig. 4. The architecture of change management system

5 Related Works

In this paper, a slight ontology framework SOF is presented to perform the semi-automatic change management to improve current management. Our approach introduces into Semantic Technology and the experiences of Business Rules Modeling to reconstruct SOF. The aim of it is to provide a continuing change management, which is lack in the Web Services.

In the research of Semantic Web Services and Workflow, there exist some proposals to support the dynamic service composition [7] [8] and the extensibility and flexibility are concerned [9] [10] [11], which present some overlaps with our ideas. Especially in [9], it presents a technique to generate composite services from high-level declarative descriptions and defines formal safeguards for meaningful composition through the use of rules. In [11], it takes into consideration of the underlying business process next to the data to present a three-level schema architecture for the conceptual design of dynamic web-based information systems to support workflow tasks and e-commerce transactions besides querying.

There are many graphical tools (ARIS, Adonis) which lay out a process model and draw connections among steps. These tools lack formal methods for verifying properties of processes. In contrast, our approach allows users to formally specify consistency constraints. Ontologies and rules are used to represent this kind of background knowledge or user’s needs. Moreover, our system can check the service profile and propose suggestions for resolving the problems.

6 Conclusions

E-Government systems are subject to a continual change. The importance of better change management is nowadays more important due to the evolution of E-Government services. It is clear that the management of changes in E-Government

will be a work which must be treated in a more systematic way in order to avoid drawbacks in the long run.

In this paper we present an approach for ontology-based change management. Our approach goes beyond a standard change management process. It is a continual improvement process. The novelty of the approach lies in the using of formal methods for achieving consistency when a problem is discovered and the formal verification of the service description.

In the future we will continue to improve this system in the following parts: Firstly, in order to improve the efficiency of this system, the system implementation and experimental evaluation will be performed. Secondly, we will extend our approach by adding function of change suggestion which can improve services by monitoring the execution of them and taking into consideration of advices from managers or end-users.

References

1. BPEL4WS. <http://www-128.ibm.com/developerworks/cn/webservices/ws-theme/ws-bpel/>
2. Maedche, A.: Managing multiple and distributed ontologies on the Semantic Web. *The VLDB Journal*, Vol. 12. (2003) 286–302.
3. OWL-S (2005). <http://www.w3.org/Submission/OWL-S/>
4. Gangemi, A.: Some Ontological Tools to Support Legal Regulatory Compliance - with a Case Study. *Workshop on Regulatory Ontologies, OTM'03*. (2003) 607–620.
5. Stojanovic, L., Abecker A.: The role of semantics in e-government service model verification and evolution. *Semantic Web meets eGovernment, AAAI Spring Symposia, AAAI*. (2006).
6. Maedche, A.: Managing multiple and distributed ontologies on the Semantic Web. *The VLDB Journal- Special Issue on Semantic Web*, Vol 12 (2003) 286–302.
7. Arpinar, I.B.: Ontology-driven Web services composition Platform. *Information Systems and E-Business Management*, Vol. 3. Springer Berlin (2005) 175–199.
8. Medjahed, B.: Composing Web services on the Semantic Web. *The VLDB Journal*, Vol. 12. Springer Berlin (2003) 333–351.
9. Medjahed, B.: Composing Web services on the Semantic Web. *The VLDB Journal*, Vol. 12. Springer Berlin (2003) 333–351.
10. Duke, A.: Enabling a callable service –oriented architecture with semantic Web Services. *BT Technology Journal*, Vol. 23. Springer Netherlands (2005) 191–201.
11. Preuner, G.: A three-level schema architecture for the conceptual design of web-based information systems: from web-data management to integrated web-data and web-process management. *World Wide Web*, Vol. 3. Springer Netherlands (2000) 125–138.