Towards Ontology-based Services Interfaces in Services Systems based Environment

Amit Bhandari
Department of CSE
RIMT-MAEC
Mandi Gobindgarh, India

Manpreet Singh
UCOE
Punjabi University
Patiala, India

ABSTRACT

Web services ontologies formally represent the knowledge and define helps in information processing with the help of concepts, relations, axioms etc defined in them. This paper outlines various services interfaces in services systems with intent to describe them ontologically. We identify two types of services interfaces in the system discussed in this paper and propose semantic-based model for the ontologies and services identified in the system differently for both the kinds of interfaces. As we delve deeper in the interfaces, we also identify that we need a state-of-the-art access control framework for both the interfaces and present a brief discussion on currently available approaches for the same. We aim to describe the system completely using WSML, so that we can look on various future aspects of the research problem like formal specification, verification, and development.

General Terms

Semantic Web Services

Keywords

SWS, WSMO, WSML, Web Services.

1. INTRODUCTION

In the last few years, various models of services providers have been identified; which includes shared services, cloud-services collectively referred to as XaaS (which includes IaaS, PaaS, SaaS, and other flavors of services offerings), and agency based services. All these kind of systems involve key-item, which is a web-service. So, a communication can be of two kinds as discussed in [4], U-S or S-S. The U-S communication refers to an interface between a user-and-a-service; whereas an S-S communication refers to an interface between service-and-a-service. For both the kinds of communications, the web service needs to be described which is the aim of this paper. A web service is a remote procedure call over http-channel in a simple language. Because of its loosely-coupled nature, it is required to implement a language-independent implementation of specifications of a web-service. To specify a web-service implementation in a language-independent schema, WSDL is used which describes various messages or function-calls implemented by a WS. Also, for the messages sent over http-channel, xml-based protocol SOAP is used which is independent of any platform-bindings and carries function-calls along with parameters wrapped in XML format. The WSDL 2.0 specifications support GET and PUT commands of HTTP protocol; hence REST is also supported by web-services [1].

The semantic web is a group of methods and technologies to allow machines to understand the meaning of information on the web [2]. Various technologies that are a part of semantic web are RDF, RDFS, OWL, and WSML. All these technologies help formally specify and describe a WS-System. The semantic web service can be defined as the semantics of the web-services to understand the meaning. The Choreography is concerned with describing U-S interfaces; and the Orchestration is concerned with describing the S-S interfaces. The language used for both choreography and orchestration is WSML for WS-System under study in this paper. The two kinds of communication interfaces identified by us are described using an example of social networking portal. The web-services in the system are identified and both U-S and S-S interfaces are focused while describing the system [5-7]. Also, we identify that there may be a given time of access to the portal for which it should become active otherwise during office hours; the access to the portal should be denied. To achieve the above effect, the access-service is given WS-Policy [3] behaviors so that it can act according to the given requirement.

The organization of the paper is as follows. Section 2 of this paper describes the related work. Section 3 and 4 discusses the system and the ontology modeling of the system. Section 5 presents the conclusion and the future scope of the research.

2. RELATED WORK

Mathes et al [8] extends the WS-Policy framework to specify validity of the services in duration. The dynamic temporal properties of the services are highlighted using various elements like startTime, endTime, expires. Further, an execution environment is setup to execute the temporal policies along with various events/actions attached to them.

The BPEL based approach described by Kallel et al [9] specify the temporal properties of web-services using XTUS-automata. The approach allows dynamic changes of temporal properties at runtime. The runtime monitoring of temporal constraints is done with the help of AO4BPEL aspects. The code generated for the temporal specifications is modular. Heinzl et al [10] uses temporal characteristics of the web services using temporal policy language to dynamically set prices according to the time of day. The implementation of the execution environment for temporal policies is done using Globus Toolkit and Apache Axis. The temporal policies have extended uses in web services, like, transfer of software licenses, account related data transfer etc.

3. SCENARIO

We consider a hypothetical enterprise OOSE Limited which has deployed enterprise wide Microsoft Office Web Apps [12]. The organization uses the standard deployment of Office Web Services (OWS) which comes along with SharePoint Services (SPS). In addition to it, the enterprise has deployed web-services which contain application-logic and policies for the OOSE only; we refer to those web-services as SEWS. The SEWS contains policies for internet access collectively categorized as temporal access characteristics for all the groups of employees of the enterprise. The employees of the enterprise are allowed to access OWS and SPS during office hours; however, for those employees who work after office hours, they are given restriction free internet access to the social networking sites like Orkut, Facebook, etc. But during office hours, there is restriction on access of these social networking sites. The administrators have chosen to use temporal policies based solution for the situation and not to filter the network traffic using some internet filtering software. The overview of the services is as visualized in figure-1.

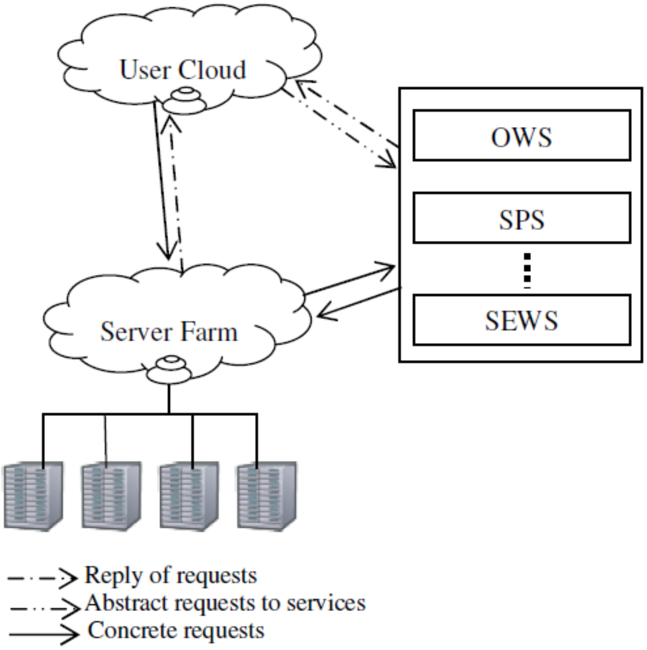

Fig 1: Overview of services present in the Enterprise

The temporal characteristics of the web services are specified using extended WS-Policy framework which uses various elements for specifying temporal policy for a web-service. The elements identified to be used for specifying the policy is; startTime, endTime, duration, timestamp. The startTime signifies the starting time for the web-service; the endTime signifies the ending time for the web-service; the duration is the total activation time in seconds; and the timestamp is the date and time when the temporal policy of the web-service was written. A sample of temporal policy for web-service should be given as in figure-2.

The web-services identified in Office Web Apps include web-based versions of Microsoft Word, Excel, PowerPoint, and OneNote [11]. The applications can be used on any browser since the support has been extended to Google Chrome, Mozilla Firefox, Safari, and Internet Explorer 7 and later versions. The OWS and SPS is basically a collection of services which include Word Viewing Service, PowerPoint Service, and Excel Calculation Service hosted in SharePoint services [12]. The figure-3 shows a detailed architecture of various services interfaces of office services in the enterprise.

Fig 2: Example of temporal policy for a web-service

```
<wsp:Policy
  Xmlns:wsp=”http://www.oose.org/temporal”
  xmlns:wsp="http://www.w3.org/ns/ws-policy" >
<wsp:ExactlyOne>
  <wsp:All>
    <tl:StartDate>01/01/09</tl:StartDate>
    <tl:EndDate>01/01/09</tl:EndDate>
    <tl:Duration>21600</tl:Duration>
    <tl:timestamp>
      01/01/09 07:01:12 AM
     </tl:timestamp>
  </wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
```

In figure-3, the three services Word Viewing Service (WVS), PowerPoint (PS), and Excel Calculation Service (ECS) are hosted in SPS. The applications Word (W), Excel (E), PowerPoint (PP), and OneNote (ON) communicate with these services on behalf of user. Hence, there exists a U-S interface between these services and the user. Also, the services mutually use functions or application logic of each other. For example, the WVS may request ECS to calculate a formula before rendering a document to the user, so, there also exist S-S interface between them when these services communicate with each other.

The SEWS services communicate with the OWS and SPS and apply temporal policies for access to the services. The temporal characteristics have been applied using extended WS-Policy model as discussed above. The services present in SEWS is as shown in figure-4. The temporal policy service (TPS) covers the temporal characteristics of the enterprise-wide services. It also contains database of the policies for each of the existing web-services on the enterprise. The connector service (CS) acts an intermediary between the TPS and every other service that exists on OOSE domain so that the temporal policy can be applied on that particular service.

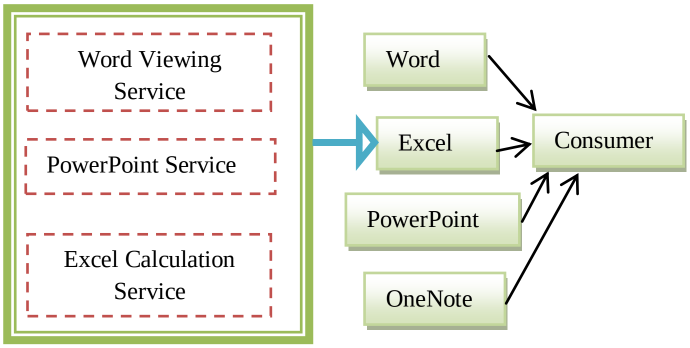

Fig 3: Detailed architecture of services interfaces of office services in OOSE

For the above described services architecture, also, there is a necessity for an access control framework which should be deployed enterprise wide so that only the authorized users get access to the services. For example, the enterprise may explore convergence of technologies in the above given scenario for which the above may appear meager. The technology convergence may be expanding of the server farm to some SaaS deployment and allowing mobile devices to have access to the services from anywhere in the world on the go. This way the enterprise may achieve the employee mobility and performance but also introduces foreign entrants to their network which was previously very well managed without any need of such an access control. But introducing the convergence of technologies, also introduces the concept of identity as to *who* is currently having access to *what* services provided by the enterprise. The SaaS convergence is collectively represented in figure-5 as a cloud ; and a brief overview of such a scenario along with access control framework is as depicted in the figure.

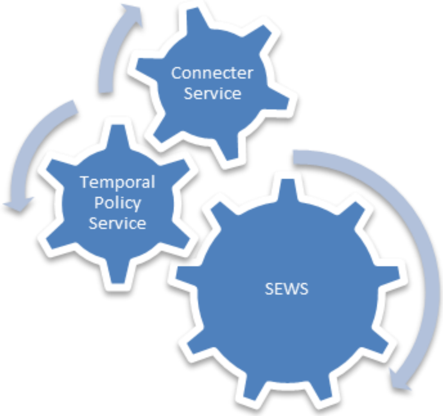

Fig 4: Services of SEWS

The access control requirement in the enterprise can be fulfilled by deploying any of the policy-based authentication and authorization frameworks like SAML [13, 14], XACML [15] or some other rule-based access control framework. Also, identity based open-framework like OAuth [16], OpenId [17] can also be deployed to fulfill the requirement of access control framework in the current scenario.

The collaboration of office services in the scenario presented above can be modeled using Web Services Modeling Language (WSML) [18]. We've used Web Services Modeling Toolkit (WSMT) [19, 20] as the modeling environment for all the services and ontology identified in the system and is discussed in the next section.

4. ONTOLOGY MODELING

The composite services identified in the previous section are

- OWS
- SPS, and
- SEWS

As, the above listed services are composite web-services, i.e., they are further composed of sub-services which perform on single type office documents like OWS consists of WVS, ECS, PS, PBS, and VS. The WVS is responsible for word documents only, ECS is responsible for excel related formula calculation and spreadsheet functions, PS is responsible for power-point related formatting and slideshow related functions, PBS is responsible for broadcast of the power-point slides, and VS is responsible of visio-related actions. Figure-6 below shows the wsml for services identified in the system.

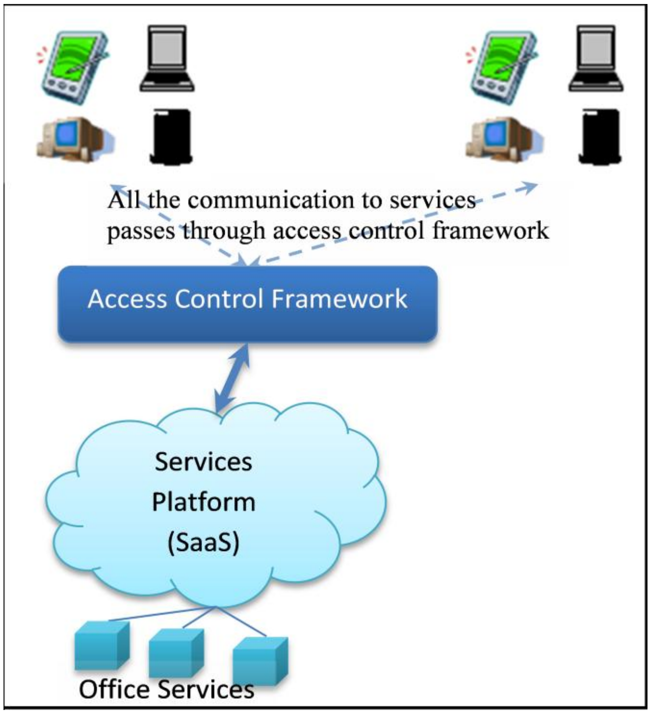
All the communication to services passes through access control framework
Access Control Framework
Services Platform (SaaS)
Office Services

Fig 5: Scenario of services with convergence of technology

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-
syntax/wsml-rule"
namespace { _"http://oose.org/services/"
, discovery
_"http://wiki.wsmx.org/index.php?title=DiscoveryO
ntology#" }

webService WVS

capability WVSCapability
     nonFunctionalProperties
          discovery#discoveryStrategy hasValue
discovery#LightweightRuleDiscovery
     endNonFunctionalProperties

effect RenderWordDocument

interface WordDocument
choreography WVSUserInterface
orchestration ExcelCalcService

interface HostDocumentService

interface EmbeddedDocCompositeService
orchestration PowePointService
```

Fig 6: WSML for WVS service

Since, SPS is an encapsulator service, which means it acts as a host for various collaborated services in OWS and also offers the traditional share-point service to the user, so, it basically consumes the OWS and offers additional set of functionality on the top of OWS so that the end-user can access the Office Web Apps through a web-browser. The SEWS is an enterprise-wide service which further hosts SPS and offers additional functionality of temporal policy for the various services and the connecter service; which helps to connect to various other services in collaborated environment. The WSML for the whole system is given in Appendix I & II.

5. CONCLUSION AND FUTURE SCOPE

A system description of services of the enterprise, named OOSE is presented in this paper. We've considered the temporal policies that need to be deployed for enforcing simplified time-based access to services. The temporal policies are applied using the extended WS-Policy framework. The two kinds of services interfaces have been identified in the system, viz., User-Service (U-S) and Service-Service (S-S). The web service modeling is done for both the kinds of interfaces using web services modeling toolkit (WSMT).

The modeling of the environment presented in the paper gives the foundation for the formal specification, development and implementation of the system under study.

6. REFERENCES

[1] Chinnici, R., Moreau, J., Ryman, A., and Weerawarana, S. 2007 Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation.

[2] Wikipedia Contributors. 2010 Semantic Web, Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Semantic_Web&oldid=406233739

[3] Vedamuthu, A., Orchard, D., Hirsch, F., Hondo, M., Yendluri, P., Boubez, T., and Yalcinalp, U. 2007 Web Services Policy 1.5 – Framework, W3C Recommendation.

[4] Bhandari, A., and Singh, M. 2011 Formal Description of Service Interfaces of SMAL Services in Services Computing Environment, 2011 International Conference on Network Communication and Computer, to appear.

[5] Pietro, I. D., Pagliarecci, F., Spalazzi, L., Marconi, A., and Pistore, M. 2008 Semantic Web Service Selection at the Process-level: the eBay/Amazon/PayPal Case Study, In Proceedins of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 605-611.

[6] Moo Nam Ko, Cheek, G. P., Shahab, M., and Sandhu, R. 2010 Social Network Connect Services, IEEE Computer, 43(8), 37-43.

[7] Wang Bin, Huang He Yuan, Liu Xiao Xi, and Xu Jing Min. 2009 Open Identity Management Framework for SaaS Ecosystem, In Proceedings of the 2009 IEEE International Conference on e-Business Engineering, 512-517.

[8] Mathes, M., Heinzl, S., and Freisleben, B. 2008. WS-TemporalPolicy: A WS-Policy Extension for Describing Service Properties with Time Constraints, In Proceedings of the 32nd Annual IEEE International Computer Software and Applications, 1180-1186.

[9] Kallel, S., Charfi, A., Dinkelaker, T., Mezini, M., and Jmaiel, M. 2009 Specifying and Monitoring Temporal Properties in Web Services Compositions, In Proceedings of the Seventh IEEE European Conference on Web Services, 148-157.

[10] Heinzl, S., Seiler, D., Juhnke, E., and Freisleben, B. 2009 Exposing Validity Periods of Prices for Resource Consumption to Web Service Users via Temporal Policies, In Proceedings of the 11th International Conference on Information Integration and Web-based Application & Services, 235-242.

[11] Neol, M., and Spence, C. 2011 Microsoft SharePoint 2010 Unleashed, Pearson Education.

[12] _____. 2010 Office Web Apps (Installed on SharePoint 2010 Products), Microsoft Corporation, http://technet.microsoft.com/en-us/library/ee855124.aspx

[13] Lockhart, H. 2005 Demystifying SAML, http://www.oracle.com/technetwork/articles/entarch/saml-084342.html

[14] Cantor, S., Kemp, J., Philpott, R., and Maler, E. 2009 Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0

[15] Rissanen, E. (ed) 2010 eXtensible Access Control Markup Language (XACML) Version 3.0, Committee Specification 01.

[16] Hammer-Lahav, E. (ed) 2010 The OAuth 1.0 Protocol, Internet Engineering Task Force (IETF), RFC:5849.

[17] Recordon, D. and Fitzpatrick, B. 2006 OpenID Authentication 1.1, http://openid.net/specs/openid-authentication-1_1.txt

[18] Jos de Bruijn, Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., and Fensel, D. 2005 The Web Service Modeling Language WSML, WSML Final Draft.

[19] Kerrigan, M. 2009 Getting Started with WSMT – Engineering and Testing Web Services and Goals, http://wiki.sti2.at/index.php?title=Getting_started_with_WSMT__Engineering_and_Testing_Web_Services_and_Goals

[20] Kerrigan, M. 2009 The Web Services Modeling Toolkit (WSMT), http://wsmt.sourceforge.net/

Appendix-I

The WSML for OfficeOntology is listed below:

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
namespace { _"http://oose.org/ows/ontologies/" }
ontology OfficeOntology

concept SharePointAppPool
        appPool ofType SharePointAppPool

concept User
        userName ofType _string
        firstName ofType _string
        lastName ofType _string
        encryptedPassword ofType _string

concept WordViewer
        useExistingAppPool ofType SharePointAppPool
        hasNewAppPool ofType _boolean
        newAppPool ofType SharePointAppPool
        configuredBy ofType User
        isOpenXMLEnabled ofType _boolean

concept PowerPointViewer
        useExistingAppPool ofType SharePointAppPool
        hasNewAppPool ofType _boolean
        newAppPool ofType SharePointAppPool
        configuredBy ofType User

concept ExcelCalculation
        useExistingAppPool ofType SharePointAppPool
        hasNewAppPool ofType _boolean
        newAppPool ofType SharePointAppPool
        configuredBy ofType User

concept ProductActivation
        hasProdKey ofType (1 *) _string
        hasInstallLoc ofType _string
        hasSharePointSettingsConfig ofType SharePointConfig

concept SharePointConfig
        isWebAppActive ofType _boolean
        hasTopLevelSite ofType _boolean
        collectionTopLevelSite ofType _string
```

Appendix-II

The WSML for identified services is given below:

1. ECS Services

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
namespace { _"http://oose.org/services/"
, discovery _"http://wiki.wsmx.org/index.php?title=DiscoveryOntology#"
, _"http://oose.org/ows/ontologies/" }

webService ECS

capability ECSCapability
     nonFunctionalProperties
          discovery#discoveryStrategy hasValue discovery#LightweightRuleDiscovery
     endNonFunctionalProperties

effect RenderExcelDocument

effect RenderSharepointList

interface PowerShell
choreography stsadmTool

interface ExcelDocument
choreography RestApiUI
orchestration JsomSvcInterface

interface ECSConfig
```

2. PowerPoint Services

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
namespace { _"http://oose.org/services/"
, discovery _"http://wiki.wsmx.org/index.php?title=DiscoveryOntology#"}

webService PS
```

```
capability PSCapability
     nonFunctionalProperties
          discovery#discoveryStrategy hasValue discovery#LightweightRuleDiscovery
     endNonFunctionalProperties

effect BrowserSlideShowInstance

effect BroadcastedSlideShowInstance

interface PBSInstance
orchestration RemoteUI

interface ImageRenderer
orchestration XAMLView

interface PowerPointDocument
choreography ImageRendererInstance
```