Inf. Sci. Lett. 1 No. 1, 49-59 (2012)

Information Sciences Letters
An International Journal

A Semantic Methodology for Customized Healthcare Information Provision

Saman Iftikhar[1], Farooq Ahmad[1] and Kiran Fatima[2]

[1]*Department of Computing, School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan*

[2]*Department of Computer Sciences, National University of Computer and Emerging Sciences, Islamabad, Pakistan*

E-mail: {saman.iftikhar, farooq.ahmad}@seecs.edu.pk, Kiranfatima_54@hotmail.com

Abstract— In healthcare domain there is a plethora of health information but no mechanism of satisfying customized users' requirements and providing demand oriented information. Healthcare domain is facing challenge of finding healthcare information that meets user requirements. Semantics can provide interoperable, automated and seamlessly meaningful communication for healthcare domain. Healthcare information is increasing with the increase of medical databases over the internet. In this matter provision of right information to appropriate users over the web is a great challenge. Semantic web technologies can address this issue by ensuring extensible, flexible and efficient information discovery related to diseases and clinical processes. We implemented a semantic registry that is a service registry as well as metadata registry. It manages and secures healthcare information and processes in a consistent way in compliance with emerging international standards and technical specifications. We evaluated our system through Patient Registration Scenario implemented semantically in our OWL-S framework.

Keywords- Semantic Registry; OWL-S; WSMO; healthcare information provision; Goal oriented discovery; HL7 Service Functional Model; Healthcare Service Ontology

I. INTRODUCTION

The advancement in Information Technology is playing increasing role in healthcare and has managed to improve the efficiency of health services to common people. Health informatics plays a vital role in the integration of Information Technology in healthcare domain. However healthcare organizations are facing problems related to communication of right information to appropriate. Due to data deluge, information retrieval and analysis has become an important problem in various fields including healthcare. Semantic web technologies provide extensible, flexible and efficient information.

The innovation and the standardization of web services provide basic building blocks for information exchange. To exploit web services to their full potential, semantics must be specified. Semantic web technologies play a pivotal role in bringing automation in the process flows. OWL-S provides ontologies for describing web services with the help of semantic constructs in an unambiguous and machine interpretable form. OWL-S follows layered structure of markup languages as HTML, XML, RDF and has built on OWL recommendation of W3C. Its ontologies describe domain concepts of services (e.g., travel, e-business, healthcare information) and business logic. The data flow and controls of the services are related to the domain ontologies through inputs, outputs, preconditions and effects. OWL-S ontologies divide service descriptions in four main parts: process model, service profile, service grounding and the service.

Currently WSMX framework provides automatic service discovery, composition and execution of web services. It provides information exchange between users and service providers and fulfill user specified goal by invoking end point web services. The main strength of WSMO over other semantic web technologies is its discovery mechanism. WSMO is based on the Web Service Modeling Framework (WSMF). WSML is used to describe services' description into Ontologies, Web services, Goals and Mediators [5].

The innovation and the standardization of web services have set the concept of web services as the basic building blocks of information technology systems for Service Oriented Architectures (SOA) applications. The idea is to explore such sophisticated SOA technologies that make the discovery of services for requested users appropriate [1]. SOA ensures interoperability, flexibility and extensibility across heterogeneous environments. In service oriented computing services are used to develop fast, economical, interoperable, evolvable, and extremely distributed applications. Services are self-governing, platform-independent entities that can be described, published, discovered, and loosely coupled [2].

Traditional approaches to services publication and discovery have generally relied on the existence of pre-defined registry services like Universal Description, Discovery and Integration (UDDI) [4]. Often the description of a service is limited in existing registry, with little or no support for problem specific descriptions. Semantic registries with the use of OWL-S attempt to overcome this limitation and provide a rich semantic description based on ontologies. Semantic matchmaking generally focuses on the problem of identifying services on the basis of the capabilities that they provide.

We proposed an OWL-S based Semantic Registry for healthcare information provision. This paper also presents healthcare service ontology [7] developed through the specifications of HL7 Service Functional Model [6], which is used in our Semantic Registry for publishing and discovering HL7 compliant healthcare semantic web services. HL7 is a well-known healthcare standard that provides specification for standardization of information exchanged among healthcare applications.

The rest of the paper is structured as follows. Section 2 explains in detail some of the related efforts. Section 3 presents the system design of our proposed approach. Section 4 discusses the detailed design. Section 5 discusses the implementation and evaluation of the proposed design. Section 6 discusses the whole idea and section 7 is the conclusions.

II. RELATED EFFORTS

Registries are important in a large scale, distributed environment, such as the semantic web. They provide the necessary functionality that allows service providers to expose information of their services to potential users. Various types of approaches that are being followed for storing and accessing information over the web are registry-based discovery mechanisms [3], indexing methods (UDDI) [4] and publish/subscribe approach [8]. In healthcare domain there is no such mechanism of binding healthcare service providers and requesters in order to discover healthcare data for use in emergency situation. There is lack of registries that provide publish and retrieval of healthcare data through web services. There is no any healthcare services publish in a semantic way for the interoperability of health information exchanged in an efficient manner. Healthcare information is more complex and has diverse dimensions. UDDI [9, 10] and ebXML [11] do not provide such semantic interoperability in healthcare domain.

In paper [9], Authors have proposed OWL-S/UDDI Matchmaker as an extension of UDDI. Before registering OWL-S based Web Services on UDDI, OWL-S/UDDI Matchmaker converts service profile of these services to UDDI data structure and then stores them on UDDI. During web service discovery, OWL-S/UDDI Matchmaker translates the services back into OWL-S format. Matching takes place between the service request and the published services advertisements present in the registry. The proposed solution enhances the UDDI registry for semantic based searching and capability based matching. UDDI registry has some inherent limitations including lack of semantic representations of contents. The matching process proposed in this paper is restricted to Inputs and Outputs matching of the service profile.

The DAML-S Matchmaker [10] was developed by the Intelligent Software Agents Group at Carnegie- Mellon University. The matchmaking system is a database where service providers can register their Web services via DAML-S descriptions through a Web interface. The system then allows service requesters to upload their service requests. The matchmaking algorithm matches the types associated with each input or output parameter. For each parameter (either input or output) there are several degrees of matching, depending on the semantic relationship between the parameters of the advertisement and the request. Based on these results a global matching result is determined.

ebXML Registry [11] give industry groups and enterprises the ability to share business semantic information and business process interfaces in form of XML. This registry has some extensions for medical data registration, annotation, discovery and retrieval in form or archetypes data definitions where registry semantic constructs are used. They provide archetype metadata ontology and describe the techniques to access archetype semantics through ebXML query facilities. They also provide mechanism, how archetype data can be retrieved from underlying clinical information systems by using ebXML Web services.

The FUSION Semantic Registry [12] is a semantically-enhanced Web service registry based on UDDI, SAWSDL and OWL. This registry augments and enhances the discovery facilities of typical UDDI registry and based on UDDI without changing its implementation. This registry performs matchmaking at data-level and developed by SEERC in the context of research project FUSION and released as open source software. Fusion registry has no matchmaking based on inputs, outputs, preconditions and effects capabilities of services.

Artemis project [13], exploits ontologies based on the domain knowledge exposed by the healthcare information standards like HL7, CEN TC251, ISO TC215 and GEHR. Artemis Web service architecture has no any globally agreed ontologies; rather healthcare institutes resolve their semantic differences through a mediator component. The mediator component works in a P2P manner and uses ontologies in order to facilitate semantic negotiation among involved institutes.

CASCOM is an agent-based approach used for semantic service discovery and coordination in mobile eHealth environment [14].

Cesar Caceres is another approach that focuses on Agent-Based Semantic Service Discovery for medical-emergency management [15].

COCOON Glue is a prototype of WSMO Discovery engine for the healthcare field to find out the most appropriate advice services [16, 17].

III. PROPOSED SYSTEM

We proposed a framework based on OWL-S semantic layer which would provide automatic service discovery, composition, invocation and execution of web services for healthcare service providers and end users. Our proposed Semantic Registry would be the key foundation block upon which electronic information is exchanged in an interoperable manner among disparate communities through web services semantics. It would be an Ontology based semantic description model explicitly represents information semantics in abstract and concrete level and resolve heterogeneity

The proposed system will consist of entry points for the communication to take place. The OWL-S/WSDL grounding mechanism would be used for end point service invocation. In our framework, we proposed to perform goal-oriented discovery with semantic matchmaking of OWL-S ontologies. The proposed use of semantic web services specification language such as OWL-S for describing web services semantically would result in better information exchange. We will incorporate three views of services into user demand to satisfy the requirements of end users in healthcare domain. These views are: customization (who is demanding information), situation (when and where the demand is occurred) and quality (how important the demand is). Service provider, who is going to provide their services for use by appropriate users, will take advantage of the complementary strengths of OWL-S, and these three views of services.

OWL-S has classes of WSDLGrounding for realizing specific elements within WSDL for OWL-S/WSDL Grounding mechanism. This mechanism is more mature as compared to WSMX. WSMX required lowering and lifting mechanism and XSLT transformations for WSMO/WSDL groundings. WSMX also uses RDF and XML as a carrier between WSML and WSDL for grounding mechanism, where loss of semantics can be observed. WSMO provides goal oriented discovery and mediation between ontologies, web services and goals that are not provided by OWL-S. In OWL-S, there is no clear distinction between choreography and orchestration. OWL-S Process Model defines choreography and orchestration. There is no need of separate management for these two processes. In WSMO, the choreography and orchestration are specified clearly. WSMX has interfaces for choreography element (provides the necessary information for communicating with the service), and the orchestration class element, (describes how the service makes use of other services in order to achieve the goal). OWL-S has no Semantic Registry for web service discovery, selection and invocation mechanism, it depends on UDDI for web services discovery. Whereas WSMX framework has three steps of discovery, Goal Discovery, Semantic Web service Discovery and End point service Discovery using any one of the approach: keyword based, light weight and heavy weight discovery [9, 10].

Our proposed framework would work with both central and distributed computing infrastructures. It will provide services for healthcare information provision and for collaboration of Personal Health Record (PHR) systems, Electronic Health Record (EHR) systems, Health Information Management (HIMS) systems, and other hospital and clinical systems.

IV. DETAILED DESIGN

Our proposed framework Fig 1 consists of following components:

- **Communication Manager**
  - Adaptor: End user request will be converted into OWL-S to make it adaptable to internal environment. OWL-S annotations of end user request will be provided to discovery component, which will perform the user oriented discovery with the help of SR, semantic matchmaking, selection & ranking components
  - **Monitoring:** This component will send the request to best selected end point web service candidate.
  - **Invoker:** The response from the end point web service will be given to the user through this component. This component will synchronize all responses from more than one end point web services.
- **OWL-S semantic layer**
  - **OWL-S ontologies:** Semantic descriptions of web services provided by service provider
  - **Semantic Registry (SR):** SR manages semantic annotations of services provided by service providers in repository and handles discovery process. It also provides OWL-S to WSDL and WSDL to OWL-S translations with the help of OWL-S ontologies.
  - **Repository:** OWL-S ontologies' semantic annotations would be stored in repository to be accessed latter for discovery purpose.
  - **Services domain ontology**
- **Discovery:** This component will perform keyword based, light weight and heavy weight discovery.
- **Semantic matchmaking:** During discovery process similar ontologies and web services will be mediated semantically.
- **Selection & Ranking:** Best candidate end point web service will be selected from the ranked list of web service.
- **Reasoner:** This component will help selection & ranking component for choosing best candidate.

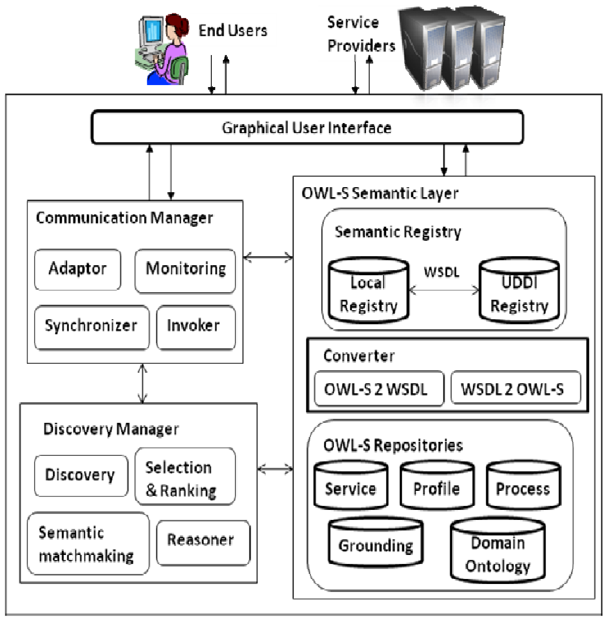

Figure 1. OWL-S based Framework

The working of the system consists of following phases:

- Information/Web services publication from service providers

- Demand oriented user discovery for healthcare information

A. Information publication

OWL-S based semantic web service is consists of three modules: Service Profile, Process Model, and Service Grounding. Service profile is used for advertisement purpose and provides data semantics. In paper [7] HL7 compliant health service capabilities were provided for publication to Semantic Registry. In order to provide functional semantics in the service description, process model is also defined for functional description of services. Fig 2 explains information publication.

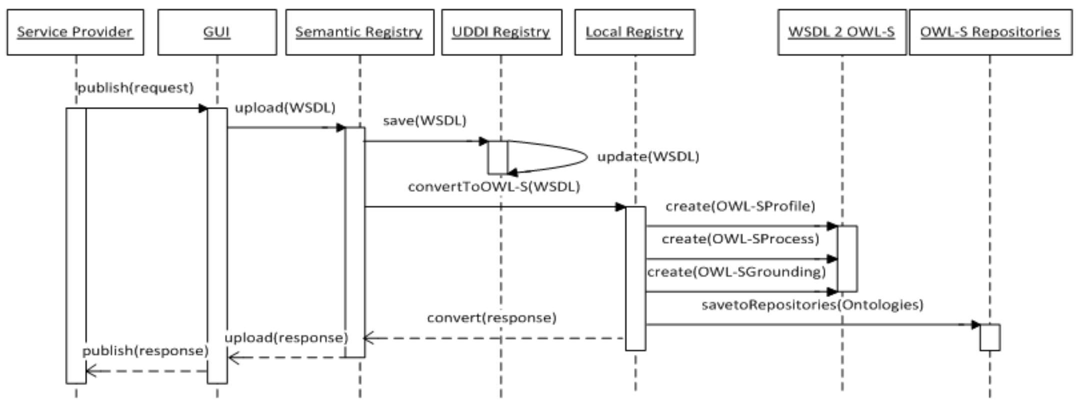

Figure 2. Healthcare information publication

B. Demand oriented information provision

The Physician and patients can now query the Semantic Registry by providing service inputs, output, preconditions or effects. As service discovery will use OWL-S service profile and service process model, the service requester have to provide the required service criteria on the basis of service inputs, output, preconditions or effects. Fig 3 explains the working of the demand oriented information provision.

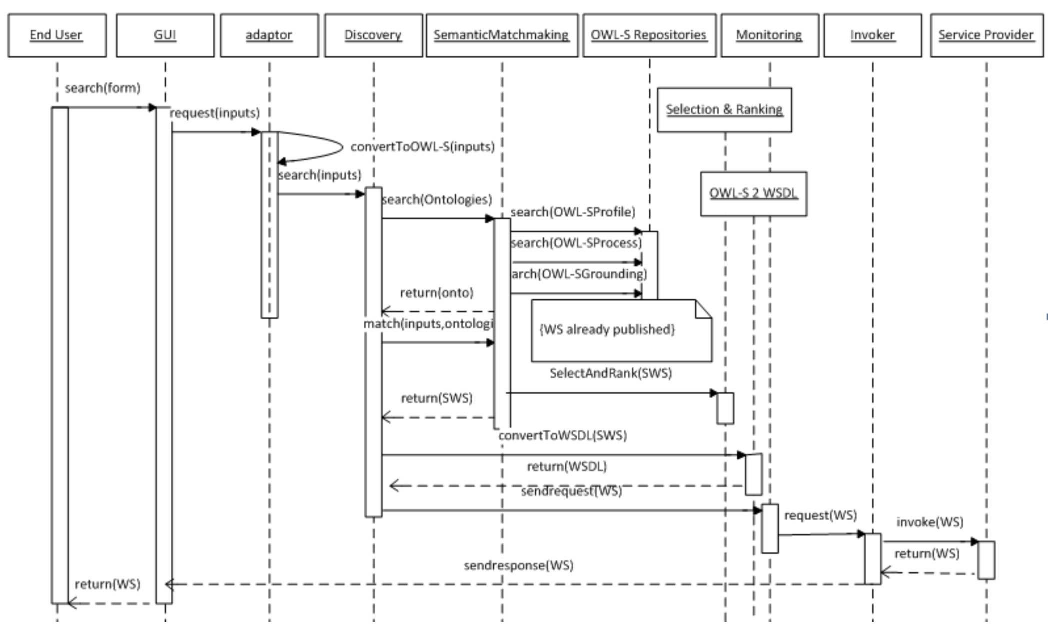

Figure 3. Healthcare information provision

V. RESULTS AND EVALUATION

How a Physician and a Patient bound on OWL-S Semantic Registry as shown in Fig 4 is better explained through a scenario. We implemented a scenario where a patient registered in a hospital through our semantic registry. The required steps for publishing and discovery phases included as follows:

- **Publishing phase**
  - 2 Web services (WSDL)
  - WSDL 2 OWL-S conversion
  - OWL-S Annotations for these 2 WSDL
    - Service, Profile, Process Model, Grounding
    - Stored in Repositories
- **Discovery phase**
  - User request
  - Convert into OWL-S
  - Map user request, OWL-S Annotations
  - Semantic matchmaking , Selection, Ranking
  - OWL-S 2 WSDL conversion
  - WSDL information from UDDI Registry
  - Service invocation and execution
  - Information provided to End user

- **Publishing phase implemented scenario**

The web services description in WSDL for Web service name addPatient is as follows:

```
<message name=”addPatient_Request”>
<part name=”Name” type=”xsd:string”>
<part name=”location” type=”xsd:string”>
</message>
<message name=”addPatient_Response”>
<part name=”patientId” type=”xs:string”>
</message>
<portType name=“addPatientPortType">
  <operation name=“addPatient">
    <input message=“addPatient: addPatient_Request "/>
    <output message="addPatient:addPatient_Response"/>
  </operation>
</portType>
<binding  name=“addPatientSoapBinding" type=“addPatient:addPatientPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name=“addPatient">
</operation>
</binding>
```

The detailed OWL-S annotations for patient registration web service are as follows:

- **Service**

```
<service:Service rdf:ID="regPatient_Service">
<service:presents rdf:resource="http://www. regPatient.com/regPatient.wsdl/regPatient_Profile#regPatient_Profile"/>
<service:describedBy rdf:resource="http://www. regPatient.com/regPatient.wsdl/regPatient_ProcessModel#regPatient_ProcessModel"/>
<service:supports rdf:resource="http://www. regPatient.com/regPatient.wsdl/regPatient_Grounding#regPatient_Grounding"/> </service:Service>
```

- **Service Profile**

```
<profile:serviceName>patientReg</profile:serviceName>
<profile:textDescription/>
<profile:hasInput rdf:resource=“ #regPatientPortType _patientReg_Name_IN"/>
<profile:hasInput rdf:resource=“#regPatientPortType _ patientReg_ Location_IN"/>
<profile:hasInput rdf:resource=“ #regPatientPortType _ patientReg_isCritical_IN"/>
<profile:hasOutput rdf:resource=“#regPatientPortType _ patientReg_patientId_OUT"/>
<profile:hasOutput rdf:resource=“#regPatientPortType _ patientReg_hospitalName_OUT"/>
<profile:hasOutput rdf:resource=“#regPatientPortType _ patientReg_contactNo_OUT"/>
 </profile:Profile>
```

- **Process Model**

```
<process:AtomicProcess rdf:ID=“#regPatientPortType _ patientReg">
<process:hasInput rdf:resource=“ #regPatientPortType _ patientReg _Name_IN"/>
<process:hasInput rdf:resource=“#regPatientPortType _ patientReg _Location_IN"/>
<process:hasInput rdf:resource=“#regPatientPortType _ patientReg_isCritical_ IN"/>
<process:hasResult> <process:Result>
<process:hasOutput rdf:resource=“=“#regPatientPortType _ patientReg_patientId_OUT"/>
<process:hasOutput rdf:resource=“=“#regPatientPortType _ patientReg _hospitalName_OUT"/>
<process:hasOutput rdf:resource=“=“#regPatientPortType _ patientReg _contactNo_OUT"/>
</process:Result> </process:hasResult>
</process:AtomicProcess>
```

- **Grounding**

```
<grounding:WsdlGrounding rdf:ID="regPatient_Grounding">
………….
rdf:ID="WSDLGrounding_regPatient_ patientReg ">
<grounding:owlsProcess rdf:resource="http://www. regPatient.com/regPatient.wsdl/regPatient_ProcessModel#regPatientPortType_ patientReg "/>
<xsd:uriReference rdf:value="http://www.regPatient.com/regPatient.wsdl# patientReg"/>
<grounding:wsdlInputMessage> // inputs
<xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#patientReg_Request"/>
……………………………
     <xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#Name"/>
    ……………………………..
     <xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#Location"/>
 ………………..
     <xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#isCritical"/>  // Outputs
     …………….
     <xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#hospitalName"/>
…………………..
 <xsd:uriReference rdf:value="http://www. regPatient.com/regPatient.wsdl#contactNo”/> </grounding:wsdlOutputMessageParts>
```

- **Discovery phase implemented scenario**
  - User request (goal oriented request)
    - Inputs: saman, seecs, true
    - Output: patient id, hospital name, contact no

- Discovered OWL-S Profile

```
<profile:serviceName> </profile:serviceName>
   <profile:textDescription/>
  <profile:hasInput rdf:resource=“ saman"/>
   <profile:hasInput rdf:resource=seecs”/>
<profile:hasInput rdf:resource=“ true"/>
   <profile:hasOutput rdf:resource=“patient Id"/>
  <profile:hasOutput rdf:resource=“hospital name"/>
  <profile:hasOutput rdf:resource=“contact no"/>  </profile:Profile>
```

Figure 4. A Physician and a Patient bound on OWL-S Semantic Registry WSDL specification for Patient Registration Web Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:hl7="urn:hl7-org:v3" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
name="PRPA_AR201306" targetNamespace="urn:hl7-org:v3" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:fi="http://java.sun.com/xml/ns/wsit/2006/09/policy/fastinfoset/service"
xmlns:tcp="http://java.sun.com/xml/ns/wsit/2006/09/policy/soaptcp/service" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702" xmlns:sc="http://schemas.sun.com/2006/03/wss/server"
xmlns:wspp="http://java.sun.com/xml/ns/wsit/policy">
<wsdl:types>
                <xsd:schema targetNamespace="urn:hl7-org:v3" xmlns:tns="urn:hl7-org:v3" elementFormDefault="qualified">
                        <xsd:include schemaLocation="PRPA_IN201311UV02.xsd"/>
                        <xsd:include schemaLocation="PRPA_IN201312UV02.xsd"/>
                </xsd:schema>      </wsdl:types>
        <message name="PRPA_AR201306_AddRequest_Accepted_PRPA_IN201311">
                <part name="body" type="hl7:PRPA_IN201311UV02.MCCI_MT000100UV01.Message"/>
        </message>
        <message name="PRPA_AR201306_AddRequest_Accepted_PRPA_IN201312">
                <part name="body" type="hl7:PRPA_IN201312UV02.MCCI_MT000300UV01.Message"/>
        </message>
        <portType name="PRPA_AR201306_PortType">
                <operation name="PRPA_AR201306_AddRequest_Accepted">
                        <input name="input1" message="hl7:PRPA_AR201306_AddRequest_Accepted_PRPA_IN201311"/>
<output name="output1" message="hl7:PRPA_AR201306_AddRequest_Accepted_PRPA_IN201312"/>
                </operation>      </portType>
        <binding name="PRPA_AR201306_Binding" type="hl7:PRPA_AR201306_PortType">
         <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
                <operation name="PRPA_AR201306_AddRequest_Accepted">
                 <soap:operation soapAction="urn:hl7-org:v3/PRPA_AR201306_AddRequest_Accepted"/>
<input name="input1">
<soap:body use="literal"/>
</input>
<output name="output1">
```

```
<soap:body use="literal"/>
                </output>   </operation>
        </binding>
        <service name="PRPA_AR201306_Service">
                <port name="PRPA_AR201306_Port" binding="hl7:PRPA_AR201306_Binding">
                        <soap:address location="http://localhost:${HttpDefaultPort}/PRPA_AR201306"/>                    </port>
        </service> </definitions>
```

- **OWL-S description of Patient Registration Web Service**

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  …
<owl:Ontology rdf:about="">
   <owl:versionInfo>Version 1.0</owl:versionInfo>
    <owl:imports rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns"/>
   <owl:imports rdf:resource=" /owls11/Service.owl"/>
  </owl:Ontology>
  <service:Service rdf:ID="PRPA_AR201306_Service">
<service:presents rdf:resource="PRPA_AR201306_Profile#PRPA_AR201306_Profile"/>
<service:describedBy rdf:resource="PRPA_AR201306_ProcessModel#PRPA_AR201306_ProcessModel"/>
<service:supports rdf:resource="PRPA_AR201306_Grounding#PRPA_AR201306_Grounding"/>
  </service:Service> </rdf:RDF>
```

- **Service Profile Ontology**

```
<profile:Profile rdf:ID="PRPA_AR201306_Profile">
   <service:presentedBy rdf:resource="PRPA_AR201306_Service"/> <profile:serviceName>PRPA_AR201306</profile:serviceName>
   <profile:textDescription/>
   <profile:hasInput rdf:resource="
PRPA_AR201306_ProcessModel#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_IN"/>
<profile:hasOutput
rdf:resource="PRPA_AR201306_ProcessModel#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_OUT"/>
   <profile:qualityRating/>
  </profile:Profile>
```

- **Service Process Model Ontology**

```
<process:ProcessModel rdf:ID="PRPA_AR201306_ProcessModel">
   <process:hasProcess rdf:resource="#PRPA_AR201306_Composite"/>
   <process:describes rdf:resource=" PRPA_AR201306_Service#PRPA_AR201306_Service"/>
</process:ProcessModel>
  <process:CompositeProcess rdf:ID="PRPA_AR201306_Composite"/>
  <process:Input rdf:ID="PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_IN">
   <process:parameterName>PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_IN
   </process:parameterName>
   <process:parameterType rdf:resource="urn:hl7-org:v3#PRPA_IN201311UV02.MCCI_MT000100UV01.Message"/>   </process:Input>
  <process:Output rdf:ID="PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_OUT">
   <process:parameterName>PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_OUT
   </process:parameterName>
```

```
    <process:parameterType rdf:resource="urn:hl7-org:v3#PRPA_IN201312UV02.MCCI_MT000300UV01.Message"/>   </process:Output>
  <process:AtomicProcess rdf:ID="PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted">
    <process:hasInput rdf:resource="#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_IN"/>
    <process:hasResult>
      <process:Result>
        <process:hasOutput rdf:resource="#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_OUT"/>
      </process:Result>     </process:hasResult>
  </process:AtomicProcess>
```


- **Service Grounding Ontology**


```
<grounding:WsdlGrounding rdf:ID="PRPA_AR201306_Grounding">
<service:supportedBy
rdf:resource="http://WSDLexamples.com/PRPA_AR201306.wsdl/PRPA_AR201306_Service#PRPA_AR201306_Service"/>
<grounding:hasAtomicProcessGrounding rdf:resource="#WSDLGrounding_PRPA_AR201306_PRPA_AR201306_AddRequest_Accepted"/>
  </grounding:WsdlGrounding>
  <grounding:WsdlAtomicProcessGrounding rdf:ID="WSDLGrounding_PRPA_AR201306_PRPA_AR201306_AddRequest_Accepted">
<grounding:owlsProcess
rdf:resource="http://WSDLexamples.com/PRPA_AR201306.wsdl/PRPA_AR201306_ProcessModel#PRPA_AR201306_PortType_PRPA_AR201
306_AddRequest_Accepted"/>
    <grounding:wsdlOperation>
      <xsd:uriReference rdf:value="urn:hl7-org:v3#PRPA_AR201306_AddRequest_Accepted"/>
    </grounding:wsdlOperation>
    <grounding:wsdlInputMessage>
      <xsd:uriReference rdf:value="urn:hl7-org:v3#PRPA_AR201306_AddRequest_Accepted_PRPA_IN201311"/>
    </grounding:wsdlInputMessage>
    <grounding:wsdlInputMessageParts rdf:parseType="Collection">
      <grounding:WsdlMessageMap>
       <grounding:owlsParameter rdf:resource="
PRPA_AR201306_ProcessModel#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_IN"/>
        <grounding:wsdlMessagePart>
          <xsd:uriReference rdf:value="urn:hl7-org:v3#body"/>
        </grounding:wsdlMessagePart>
      </grounding:WsdlMessageMap>
    </grounding:wsdlInputMessageParts>
    <grounding:wsdlOutputMessage>
      <xsd:uriReference rdf:value="urn:hl7-org:v3#PRPA_AR201306_AddRequest_Accepted_PRPA_IN201312"/>
    </grounding:wsdlOutputMessage>
    <grounding:wsdlOutputMessageParts rdf:parseType="Collection">
      <grounding:WsdlMessageMap>
     <grounding:owlsParameter rdf:resource="
PRPA_AR201306_ProcessModel#PRPA_AR201306_PortType_PRPA_AR201306_AddRequest_Accepted_body_OUT"/>
        <grounding:wsdlMessagePart>
          <xsd:uriReference rdf:value="urn:hl7-org:v3#body"/>
        </grounding:wsdlMessagePart>
      </grounding:WsdlMessageMap>
    </grounding:wsdlOutputMessageParts>
```

```
  <grounding:wsdlReference>
    <xsd:uriReference rdf:value="http://www.w3.org/TR/2001/NOTE-wsdl-20010315"/>
  </grounding:wsdlReference>
</grounding:WsdlAtomicProcessGrounding>
```

VI. DISSCUSSIONS

We will provide such a semantic based registry or OWL-S framework that provides services and metadata to manage healthcare information and processes in a consistent way that would be compliant with emerging international standards. Our Solution would provide collaboration and give hospitals and clinics the ability to share healthcare semantic information. Semantic web technologies can provide extensible, flexible and efficient information. Semantics can provide interoperable, automated and seamlessly meaningful communication in healthcare domain.

VII. CONCLUSIONS

In this paper we proposed a framework for semantic registry based on OWL-S - an ontology web language for web services. We implemented a scenario where we bound a physician and a patient for registering to a hospital. The whole OWL-S description for the scenario is also explained.

REFERENCES

[1] Erl, T., (2005). “Service-Oriented Architecture (SOA). Concepts, Technology, and Design”. Prentice Hall PTR.

[2] Papazoglou, M.P. and W-J. van den Heuvel, “Service-Oriented Architectures: Approaches, Technologies and Research Issues,” VLDB J., vol. 16, no. 3, 2007, pp. 389-415.

[3] S. Willmott, H. Ronsdorf, K. Krempels, “Publish and search versus registries for semantic web service discovery”, In the proceeding of Web Intelligence, 2005.

[4] Clement, L. Hately, A. Riegen, C. and Rogers, T. “UDDI Version 3.0.2”. OASIS, 2004. UDDI Spec Technical Committee Draft. Available at http://www.uddi.org.

[5] Web Services Modeling Ontology. http://www.wsmo.org/

[6] Health Level 7. http://HL7.org

[7] Saman Iftikhar, Falak Nawaz, H. Farooq Ahmad, Kiran Fatima, "Introducing Semantics in DHTs for Grid Services in a Semantic Registry“, 6th IEEE International Conference on Emerging Technologies, pp 382 - 387, 18-19 Oct. 2010

[8] Nawaz, F., Pasha, M., Ahmad, F., Suguri, H., “Pushing Semantic Web Service Profiles to Subscribers for Efficient Service Discovery”. In the 3rd International Conference on Semantics, Knowledge and Grid (SKG ‘07) Xi’an, China, October 2007.

[9] N. Srinivasan, M. Paolucci, K. Sycara, ”An Efficient Algorithm for OWL-S based Semantic Search in UDDI”, First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004 96-110.

[10] M. Paolucci, T. Kawamura, T. Payne, K. Sycara, “Semantic Matching of Web Services Capabilities”, In the Proceedings of the First International Semantic Web Conference on The Semantic Web, p.333-347, June 09-12, 2002.

[11] Asuman Dogac, Gokce B. Laleci, Yildiray Kabak, Seda Unal, Thomas Beale, Sam Heard, Peter Elkin, Farrukh Najmi, Carl Mattocks, David Webber, “Exploiting ebXML Registry Semantic Constructs for Handling Archetype Metadata in Healthcare Informatics”. In proceeding of International Journal of Metadata, Semantics and Ontologies (IJMSO-08).

[12] Kourtesis D. and Paraskakis I. Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery. In Bechhofer S. et al.(Eds.): ESWC 2008, Lecture Notes in Computer Science 5021, Springer-Verlag Berlin Heidelberg 2008, pp. 614-628.

[13] Dogac A., Laleci G., Kirbas S., Kabak Y., Sinir S., Yildiz A., and Gurcan, Y.: Artemis: deploying semantically enriched web services in the healthcare domain information systems journal (Elsevier), 2006, 31, (4–5), pp. 321–339

[14] CASCOM: Nadine Fröhlich, Heikki Helin, Heimo Laamanen, Thorsten Möller, Thomas Schabetsberger, Heiko Schuldt, and Christian Stark, Semantic Service Co-Ordination for Emergency Assistance in Mobile e-Health Environments, Workshop on Semantic Web in Ubiquitous Healthcare, collocated with the 6th International Semantic Web Conference (ISWC2007), 2007.

[15] Cesar Cáceresc, Alberto Fernández, Sascha Ossowski, Carlos Matteo Vasirani, Agent-Based Semantic Service Discovery for Healthcare: An Organizational Approach, IEEE Intelligent Systems, vol. 21, no. 6, 2006, pp. 11-20.

[16] Cocoon: Emanuele Della Valle, and Dario Cerizza. The mediators centric approach to automatic Web Service discovery of Glue. CEURWorkshop Proceedings, vol. 168, 2005, pp. 35–50.

[17] http://www.cocoon-health.com

[18] Healthcare Services Specification Project (HSSP) HL7 Services Oriented Architecture SIG; http://hssp.wikispaces.com.

[19] Protégé API. Jena, http://jena.sourceforge.net/.