



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Ontologising Interaction Behavior for Service-Oriented Enterprise Healthcare Integration
Author(s)	Sahay, Ratnesh; Fox, Ronan; Hauswirth, Manfred
Publication Date	2008
Publication Information	Ratnesh Sahay, Ronan Fox, Manfred Hauswirth "Ontologising Interaction Behavior for Service-Oriented Enterprise Healthcare Integration", Proceedings of the 6th IEEE European Conference on Web Services (ECOWS 2008), 2008.
Publisher	IEEE
Item record	http://hdl.handle.net/10379/573

Downloaded 2023-03-27T20:02:50Z

Some rights reserved. For more information, please see the item record link above.



Ontologising Interaction Behavior For Service-Oriented Enterprise Healthcare Integration

Ratnesh Sahay, Ronan Fox and Manfred Hauswirth
Digital Enterprise Research Institute
National University of Ireland, Galway
firstname.lastname@deri.org

Abstract

In this paper we analyse the HL7 healthcare standard as an integration mechanism to connect service-oriented healthcare enterprises. Healthcare enterprises differ in their process model even if they follow same standard. This difference is due to the way in which healthcare is influenced by various stakeholders within regional clinical practices. Thus the design of the interaction behaviour i.e., HL7 interactions, of communicating healthcare enterprises is subject to local implementation. We present an example scenario that shows how heterogeneous process models evolve, even if healthcare care enterprises follow a similar standard such as HL7. We present an approach that enables the separation of the process layer from HL7 profiles to enable control and to resolve the heterogeneity of the enterprise interaction behavior. We apply semantics on top of HL7 profiles to resolve ambiguity and heterogeneity in the service and process definitions of HL7 compliant healthcare enterprises. We propose an integration platform called PPEPR: Plug and Play Electronic Patient Records, which is based on the principals of semantic Service-Oriented Architecture (sSOA).

1. Introduction

Web services provide the technological foundation for implementing and delivering SOA platforms. In recent years various research and industrial efforts have focussed on Service-Oriented Architectures (SOAs) and Web service technology. Two of the core challenges of conventional computing—search and integration—(also known as semantic gap of SOA) are not addressed by SOAs. Therefore, SOA itself is not a complete solution for the integration of service-oriented enterprise information systems and success of Web services and SOA itself still depends on resolving three fundamental challenges, namely search, inte-

gration and mediation [2, 3]. The integration and/or interoperability requirements of service-oriented enterprise information systems have resulted in the development of new types of SOAs, called semantic Service-oriented Architecture (sSOA) where this "semantic gap" is solved by applying ontologies on top of SOAs to resolve ambiguity and heterogeneity in data and service definitions. Unfortunately, this "semantic gap" is not well defined for an enterprise integration solution. This is due to the fact that the size of the gap varies between domains and depends on the particular context where SOA based enterprise solutions are deployed. A generalised approach for all enterprises is not useful as each domain has its own complexities and interoperability requirements. A domain-based, balanced approach which includes domain knowledge (e.g., simple taxonomies, ontologies) technology (e.g., Web services, semantic tools) and a domain-specific development methodology (e.g., top-down, bottom-up) is required to achieve meaningful enterprise integration solution.

We have introduced a functioning EPR (Electronic Patient Records) integration platform, called PPEPR¹: Plug and Play Electronic Patient Records. Our focus within PPEPR is on HL7² (Health Level Seven) standard, which is due to the fact that HL7 is the most widely used message based healthcare communication standard. In the HL7 Standard, there are two major versions, HL7 v2 and v3. While the HL7 v2 standard was created mostly by clinical interface specialists, the v3 standard has been influenced by medical informaticians. HL7 v2 messages are unstructured and flexible involving optional fields and segments whereas HL7 v3 is structured and provides greater consistency across the entire standard. HL7 v3 has published Web-service [12] and SOA4HL7 [7] profiles to support healthcare enterprise workflows and benefits from interoperability features offered by the Web service technologies.

¹<http://ppepr.deri.ie/>

²<http://www.hl7.org/>

Web service-enabled enterprises expose their external public behavior without revealing the internal private functionality or behavior. Enterprise services interact with each other according to their behavioral description (externally and/or internally). We describe such behavior as an interaction behavior and semantic (ontological) description of interaction behavior is described as behavioral semantics. In this paper, first we analyse the integration requirements of HL7 compliant healthcare enterprises at service and process levels. Secondly, we present an example scenario which explains how heterogeneous service and process models evolve, even if healthcare care enterprises follow a particular standard like HL7. Then, we describe how ontologies are applied on top of both the HL7 profiles to fill the "semantic gap" for service and process definitions. Finally, PPEPR's integration solution is then specified in terms of interactions between healthcare services. The major contribution of this paper is to present an approach for ontologising interaction behavior for service-oriented enterprise healthcare and to enable interaction between healthcare enterprises in the presence of heterogeneous service and process definitions.

2. HL7, SOA, and Web-Service

HL7 version 3 Web service profile provides the capability of transporting existing HL7 v3 messages using Web service protocols. The intention of this WS profile is to achieve "plug-and-play" interoperability via Web services in a healthcare environment. In this environment Independent Software Vendors (ISV) and corporate developers implementing HL7 interfaces can write generic and reusable classes, subroutines, and modules consistent with the guidelines set forth by the HL7 for Web services standards in order to handle HL7 message traffic from a potentially unlimited number of connecting applications and services. If applications that "expose" HL7 messages follow the HL7 Web services profile (WSP) guidelines, then "consumers" of HL7 messages can be written without prior knowledge of the interacting applications. Three major issues from an integration perspective are:

1. The service definition becomes superfluous, this leads to message definition approach where service clients are automatically able to interoperate based on the messaging definition.
2. The WS-profile assumes that all different healthcare entities will follow the particular standard.
3. Message, service, and process definitions are tied together. Thus, there is an absence of "separation of concern".

One major benefit of this approach is that "prior knowledge" or a single "agreed" model is not required at the communication level but still assumes a single "agreed" model at specification level where all healthcare entities should follow the Web service profile. It is common that people who manage information, most often have different ways of interpreting it. For example, most of IT or healthcare professionals interpret medical standards differently, thus challenging the purpose of industry standards [8].

HL7 version 3 SOA4HL7 profile is a guideline for implementing healthcare services within a Service Oriented Architecture. SOA4HL7 complements the Service Specification Framework (SSF) defined within the Healthcare Services Specification Project (HSSP)³, but provides an additional interim method of defining and implementing Web services based on HL7 v3 artifacts. Two major integration issues are:

1. The SOA4HL7 profile is intended to provide a top-down "service based" approach, which means that the service definition (or service contract) becomes key and needs to be available to the client at design time. This requires a single, agreed service definition model, in the form of an approved industry standard specification.
2. Even though the SOA4HL7 profile has separated the message definition from the service definition, it still lacks the separation of the "processes" from the services. As we discussed above this separation is important because each healthcare enterprise differs in their process models even if they follow the same standard. In this regard, the separation of interaction behavior, e.g., HL7 message exchange pattern⁴ within a process (orchestration and/or choreography) layer is required to control and manage the heterogeneity of interaction behavior at the separate layer.

The focus of this paper is to present an approach for the semantic description of interaction behavior called behavioral semantics. The behavioral semantics enables the interaction between healthcare enterprises in the presence of heterogeneous service and process definitions. The behavioral semantics is the formal description which defines a service's external (public) and internal (private) behavior. The external behavior describes a protocol that can be used by a client to consume the service functionality. The internal behavior describes a workflow, i.e., how the functionality of the service is aggregated out of other services [17]. In PPEPR, a behavioral ontology is being developed for the semantic description of a service's external behavior and functional ontology for the internal behavior.

³<http://hssp.wikispaces.com/>

⁴<http://www.w3.org/TR/wsdl20-adjuncts/#meps>

A detailed description of HL7 profiles is outside the scope of this paper. We have described the standard from the perspective of a service-oriented solution for enterprise health-care integration. Based on the above discussion our approach addresses the following integration requirements:

- Identifies the “semantic gap” between and within SOA, HL7 WS and SOA profiles.
- Applies ontologies (functional, behavioral) to resolve ambiguity and heterogeneity between service and process definitions.
- Provides a healthcare standard based flexible architecture that includes top-down and bottom-up development methodologies.
- Follows a semi-automatic integration approach, where ontologies (schema level) are constructed and mapped at design time to be mediated at runtime (instance level).
- Enables the “separation of concern”, between health-care services and process.

3. Behavioral Semantics in sSOA for HL7

Healthcare is a complex domain, comprising vendors, standards, legacy systems, and information systems that differ inherently from one another. PPEPR: Plug and Play Electronic Patient Records is based on the design principles of a semantic SOA Reference Architecture [18] and is built around semantic Web service technology. The PPEPR architecture considers three types of integrations between EPRs (Electronic Patient Records) systems based on their Web service capabilities (or lack thereof).

EPR (non-Web service) ↔ EPR (non-Web service):

This type of interaction is focussed on existing EPRs, which are mostly HL7 v2.x based.

EPR (non-Web service) ↔ Web Service enabled EPR:

This type of integration is the most complex (e.g. HL7 2.x ↔ HL7 v3), since EPRs (non-Web services) are required to communicate with the other EPRs (Web-services).

Web Service EPR (1) ↔ Web Service EPR (2): This type of integration in PPEPR architecture offers the best interoperability solution by achieving syntactic as well as semantic interoperability. The focus of this paper is on integration of this type.

A detailed description of PPEPR and its architecture is available in [13]. A detail description of semantic Web service technologies applied within PPEPR is outside the

scope of this paper. PPEPR’s semantic Web service technologies are based on the WSMO (Web service modeling ontology) framework [11]. For a service’s external behavior, WSMO defines a choreography distinct from WS-CDL⁵ (WS-CDL defines a common global viewpoint of the observable behavior of collaborating services whereas in WSMO the choreography and orchestration is part of the interface definition of a service description) [4]. In PPEPR, the common global viewpoint is implicit as services are based on HL7 defined message exchange patterns, i.e., HL7 interaction or storyboard, and the behavioral ontology is designed for the semantic description of message exchange patterns.

The internal behavior of a service is semantically described by a functional ontology. HL7 categorises healthcare events, i.e., HL7 trigger events, and the PPEPR functional ontology is based on this categorisation, where each HL7 trigger event is a Web service within PPEPR. A functional ontology is a semantic description of HL7 based healthcare events. HL7 v2 and v3 differ syntactically in their structure of trigger events. Therefore, functional ontologies are created and mapped based on their similarity.

To model and execute message exchange patterns, it is necessary to employ a process modelling and execution standard which is able to reference ontological elements and allow their mapping within the model. BPEL for Semantic Web Services (BPEL4SWS⁶ [6]), a set of language extensions to BPEL enables the referencing of ontological elements within a business process description. BPEL4SWS facilitates the orchestration of Semantic Web Services using a process based approach and is coupled with its ontological representation which is called sBPEL. In order to relate the semantics pertaining to one element in the BPEL4SWS description an additional attribute `modelReference` identifies the corresponding ontological instance in the sBPEL process model.

4. Example Scenario

This section presents an example scenario, which consists of six messages including request for patient’s lab test, lab test result, response, and confirmation messages [Figure 1]. The use case has three actors:

EPR System, General Practitioner (GP) This EPR is HL7 v3 compliant and it places a Lab test order fulfilment request to another independent EPR system (hospital laboratory).

EPR System, Hospital Laboratory This EPR is HL7 v2.5 and HL7 v2 Clinical Document Architecture

⁵<http://www.w3.org/TR/ws-cdl-10/>

⁶<http://www.ip-super.org/>

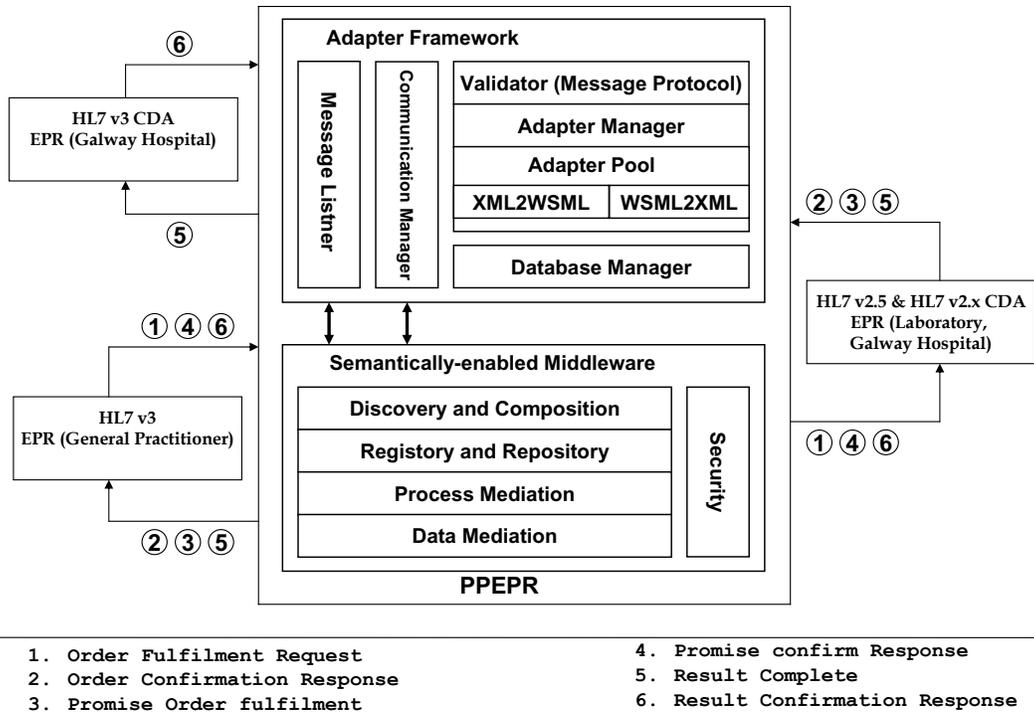


Figure 1. PPEPR Architecture and Lab Test Order Example Scenario

(CDA)⁷ compliant. The hospital Laboratory receives the order for patient’s lab test results from HL7 v3, HL7 v2.x, and HL7 v2/v3 Clinical Document Architecture (CDA) compliant EPRs.

EPR System, Galway Hospital This EPR is HL7 v3 CDA (Clinical Document Architecture) compliant and receives lab test result from HL7 v 2.x and v2/v3 Clinical Document Architecture compliant Hospital Laboratory.

Each actor has a specific ‘application role’, e.g., Order Placer as General Practitioner (GP), Order Fulfiler as Hospital Laboratory, and Result Receiver as Galway Hospital and General Practitioner (GP), and PPEPR acts as an integration platform. Based on this example scenario we describe in Section 5 how behavioral semantics are defined for both the versions (HL7 v3 and v2) and how semantic Web service and process definitions are annotated by functional and behavioral ontologies.

5. Semantics for Service & Process Definitions

Figure 2 describes the PPEPR approach for developing semantically-enabled service (WSML) and process

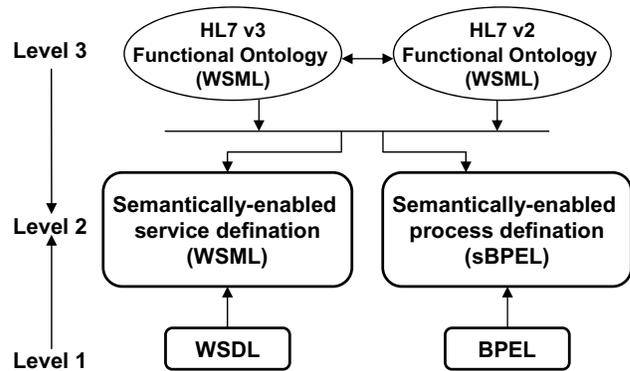


Figure 2. Ontologising Healthcare Service (WSDL) & Process definitions (BPEL)

(sBPEL) definitions. Level 3 shows the semantic descriptions (functional ontologies) of EPR services, Level 2 shows the semantic definitions of services and processes, and Level 1 is the syntactic definition of services (WSDL) and processes (BPEL). To integrate a new EPR, a semantic service definition (Level 3 → Level 2, top-down) is created first whereas existing systems are integrated in bottom-up (Level 1 → Level 2) fashion. This involves a manual process of transforming WSDL/BPEL to WSML/sBPEL. We are investigating means to incorporate the work

⁷<http://xml.coverpages.org/hl7CDA-Ann.html>

in [16] to automate the WSDL – WSMML grounding task. Grounding and invocation of services are performed by the semantically-enabled middleware (WSMX).

PPEPR’s functional and behavioral ontologies are designed to annotate service and process definitions. Figure 3

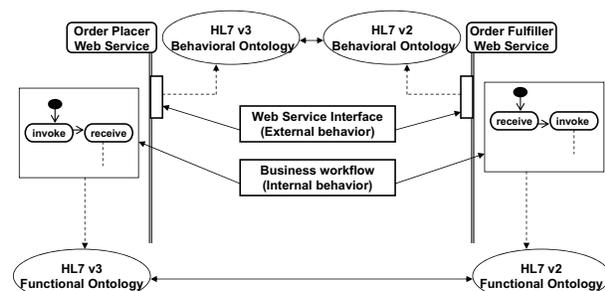


Figure 3. Semantics for Web Service's External and Internal behavior

shows how Web services (Order placer and fulfiller) interfaces and internal workflows are annotated by behavioral and functional ontologies. The bi-directional arrow indicates mapping between ontologies. Behavioral ontology provides the ontological reference to Web service interface, i.e., public external behavior. Functional ontology provides the ontological reference to internal workflows of order placer and fulfiller.

Figure 4 shows the choreography between the three actors of the PPEPR’s use case, where the Order placer (GP EPR) initiates the lab order fulfilment request. The request activates with a trigger event “Placer order activate” which maps to a similar trigger event OML^021 (in HL7v2). The order fulfiller (Lab) sends the confirmation receipt of an order followed by a trigger event (“filler promise activate(HL7v3)” or “ORU^022(HL7v2)”) that sends a promise message (which can also be rejected) to fulfill the order. The final two messages are the lab test results sent followed by the confirmation from the order receiver (Hospital EPR) and order placer (GP EPR).

Listings 1 to 5 show the snippets of the PPEPR namespaces, behavioral ontologies, mapping definition of functional ontologies and semantically enabled Web service definition.

Listing 1 describes namespace declarations of a message, functional, behavioral ontologies, grounding (WSML to WSDL), and General Practitioner (GP) EPR Web service.

```

1 namespace {
2   dc .."http://purl.org/dc/elements/1.1/",
3   xsd .."http://www.w3.org/2001/XMLSchema#",
4   LabServiceGrounding .."http://host/services/EPR_Lab.wsdl",
5   GpServiceGrounding .."http://host/services/EPR.GeneralPractitioner.wsdl"
6
7   HLv3Fontology .."http://host/HL7v3FunctionalOntology#",
8   HLv2Fontology .."http://host/HL7v2FunctionalOntology#",

```

```

8 LabEPRbehavior .."http://host/gpEPRbehavioralOntology#",
9 GpEPRbehavior .."http://host/labEPRbehavioralOntology#"
10 b4s .."http://ip-super.org/BPEL4SWS#"
11 sa .."http://www.w3.org/2002/ws/sawSDL/spec/sawSDL#"
12 HL7v2 .."http://www.hl7-v2.org#",
13 HL7v3 .."http://www.hl7-v3.org#"

```

Listing 1. Namespace declarations

Listing 2 shows the ontological description of the trigger event (PlacerOrderActivate) within choreography [Figure 4]. Both functional ontologies are imported (line 2) in case mediation is required to represent the equivalent HL7v2 trigger event(OML^021). PlacerOrderActivate is a “subConcept” (line 5) of an equivalent concept defined in the functional ontology. The behavioral ontology describes only those concepts of the functional ontology that are publicly visible and describes the state of choreography. For example, the ‘mode hasOut’ (line 9) means that it can only be changed by the GP EPR, ‘write access’ to outside environment is not permitted. A grounding mechanism (line 10) is provided that implements ‘read access’ for the outside environment. The ontology used to describe states in a WSMO choreography are extensions to the standard WSMO ontology. Attributes, e.g., ‘mode’ are defined as a non-functional properties.

```

1 ontology .."http://host/gpEPRbehavioralOntology"
2 importsOntology
3 { .."http://host/HL7v3FunctionalOntology",
4   .."http://host/HL7v2FunctionalOntology" }
5 concept PlacerOrderActivate subConceptOf
6   HLv3Fontology#PlacerOrderActivate
7   nonFunctionalProperties
8     dc#description hasValue "sends order fulfillment request"
9     mode hasValue hasOut
10    grounding hasValue GpServiceGrounding#
11      Placer_ActivateLabOrder
12    endNonFunctionalProperties
13 .

```

Listing 2. Snippet of behavioral ontology for General Practitioner (GP)

Listing 3 shows the behavioral ontology of Lab EPR (order fulfiller) and lines 6-13 semantically describes two trigger events, e.g., FillerPromiseActivate, ResultEventComplete).

```

1 ontology .."http://host/labEPRbehavioralOntology"
2 importsOntology
3 { .."http://host/HL7v3FunctionalOntology",
4   .."http://host/HL7v2FunctionalOntology" }
5 concept FillerPromiseActivate subConceptOf
6   HLv3Fontology#FillerPromiseActivate
7   nonFunctionalProperties
8     dc#description hasValue "sends promise to perform the
9     ordered service"
10    mode hasValue hasOut
11    grounding hasValue LabServiceGrounding#Filler.Promise
12  endNonFunctionalProperties
13 concept ResultEventComplete subConceptOf
14   HLv3Fontology#ResultEventComplete
15   nonFunctionalProperties

```

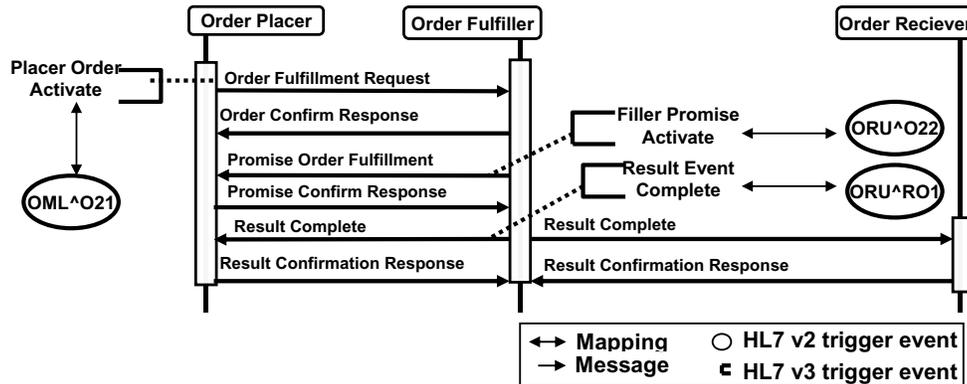


Figure 4. Choreography[Lab Test Order] between Order Placer, Order Fulfiler and Result Receiver

```

15     dc:description hasValue "Sends Result complete with lab test
16         result"
17     mode hasValue hasOut
18     grounding hasValue LabServiceGrounding#
19         Result_CompleteLabResult
20     endNonFunctionalProperties

```

Listing 3. Snippet of behavioral ontology for Laboratory

Listing 4 shows the mapping definition between two semantically similar trigger events “ResultEventComplete” of HL7v3 and ORU^R01 of HL7v2 [Figure 4]. This mapping example is in the Abstract Mapping Language (AML) [14] syntax, which is formalism-neutral, and grounded to WSM. Lines 2-5 describe source (onto1) and target (onto2) ontologies, lines 6-7 comprise a single statement that defines mapping of ‘ResultEventComplete’ to ORUR01.

Lines 8-13 describe two concepts (Class) that are mapped and finally line 14 describes the type of mapping relation, e.g., ClassMapping, AttributeMapping.

```

1 <Alignment><dc.identifier rdf:resource="v2v3F.Mapping"/>
2 <onto1><formalism name="WSML" uri="http://www.wsmo.org/wsm1"/>
3 <uri>http://dev.ppepr.der.i.e/HL7v3FunctionalOntology</uri> </onto1
4 >
5 <onto2><formalism name="WSML" uri="http://www.wsmo.org/wsm1"/>
6 <uri>http://dev.ppepr.der.i.e/HL7v2FunctionalOntology</uri> </onto2
7 >
8 <map><Cell id="http://dev.ppepr.der.i.e/HL7v3FunctionalOntology.wsm1
9 #ResultEventComplete
10 http://dev.ppepr.der.i.e/HL7v2FunctionalOntology.wsm1#ORUR01">
11 <entity1>
12 <Class rdf:about="http://dev.ppepr.der.i.e/HL7v3FunctionalOntology.wsm1
13 #ResultEventComplete">
14 </Class></entity1>
15 <entity2>
16 <Class rdf:about="http://dev.ppepr.der.i.e/HL7v2FunctionalOntology.wsm1
17 #ORUR01">
18 </Class> </entity2>
19 <relation>ClassMapping</relation> </Cell> </map>
20 </Alignment>

```

```

16 .
17 .

```

Listing 4. Mapping definition (functional ontologies) of ResultEventComplete (HL7v3) & ORUR01 (HL7v2)

Listing 5 shows the code snippet of semantic Web service definition of General Practitioner(GP)’s EPR system. The snippet only describes the structural arrangement of the WSMO-based semantically enabled service definition. Line 2 is a Web service URL of GP EPR, lines 3-6 describe the Web service as non-functional property. Lines 7-9 import all required ontologies, and lines [10-13] describes the interface (GP_Interface) of the GP EPR. It describes how to interact with a service from the requester point of view (WSMO choreography) and how the service interacts with other services.

```

1 WebService
2 _"http://host/EPR_GeneralPractitioner.wsm1"
3   nfp
4     dc:title hasValue "Electronic Patient Record for General
5       Practitioner System
6       (HL7 Order Placer or Result Receiver)"
7   endnfp
8   importsOntology {
9     _"http://host/HL7v3FunctionalOntology",
10    _"http://host/HL7v2FunctionalOntology" }
11 interface
12 _"http://host/EPR_GeneralPractitioner#GP_Interface"
13   choreography GeneralPractitionerChoreography
14   stateSignature _"http://host/gpEPRbehavioralOntology#"
15 .

```

Listing 5. WSML Web service definition for General Practitioner EPR

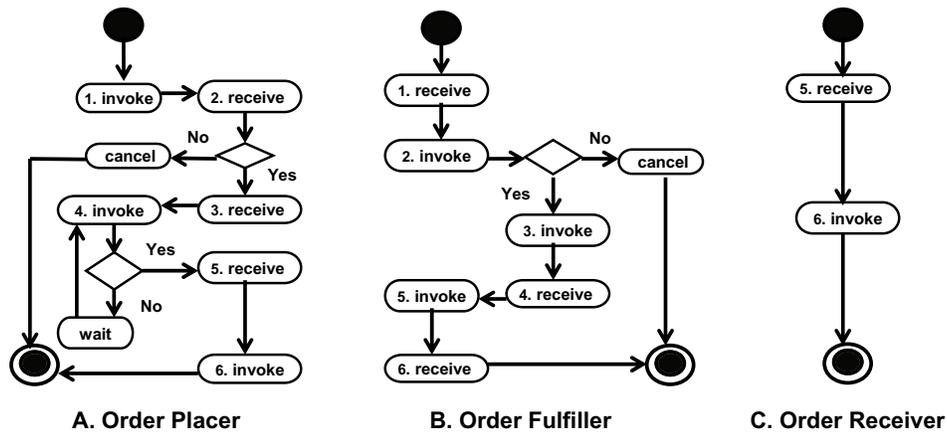


Figure 5. Business Process Models_{ABC} for Order Placer, Fulfiller, and Receiver in a HL7 Lab Test Order Request

As discussed above, HL7 not only defines the message content, but also the business logic to achieve a certain functionality in the health care domain. Figure 5 shows the process models of the order placer, fulfiller, and receiver required to achieve the actual healthcare process. It is sufficient if three actors, the process placer, the process fulfiller, and the process receiver model and execute a process according to the message exchange patterns defined in HL7 and shown in Figure 4.

```

1 <b4s:conversations>
2 <b4s:conversation b4s:name="OrderPlacer" />
3 <b4s:conversation b4s:name="OrderFulfiller"/>
4 </b4s:conversations>
5
6 <sequence>
7 <b4s:invoke name="OrderFulfillmentRequest" sa:modelReference="http
://host/HL7v3FunctionalOntology#PlacerOrderActivate"
partnerLink="OrderFulfiller" portType="spwsdl:
OrderFulfillerPortType" operation="Fulfiller_ActivateLabOrder"
variable="LabRequest" createInstance="yes" />
8
9 <extensionActivity>
10 <b4s:receive name="OrderFulfillmentRequestAck" modelReference="
http://host/HL7v3FunctionalOntology#PlacerOrderActivateAck"
b4s:inputVariable="RequestAck" b4s:outputVariable="
OrderActivateFulfillmentRequest" b4s:conversation="
FulfillmentRequest" />
11 </extensionActivity>
12 </sequence>
13 .
14 .

```

Listing 6. Snippet of BPEL4SWS Order Placer(GP)

Listings 6- 8 show how PPEPR’s functional ontologies are referenced by a BPEL4SWS documents, for example, in listing 6, lines 1-4 define a new element <b4s:conversation> introduced by BPEL4SWS.

```

1 <b4s:conversations>
2 <b4s:conversation b4s:name="OrderPlacer" />
3 <b4s:conversation b4s:name="OrderFulfiller"/>
4 <b4s:conversation b4s:name="OrderReceiver"/>
5 </b4s:conversations>
6
7 <sequence>
8 <b4s:receive name="OrderFulfillmentRequest" sa:modelReference="
http://host/HL7v2FunctionalOntology#OML021" partnerLink="
OrderPlacer" portType="spwsdl:OrderPlacerPortType" operation=
"Placer_LabOrder" variable="LabRequest" createInstance="yes"
/ >
9
10 <extensionActivity>
11 <b4s:invoke name="OrderConfirmation" modelReference="http://host/
HL7v2FunctionalOntology#OML022" b4s:inputVariable="
ConfirmationResponse" b4s:outputVariable="
OrderConfirmationResponse" b4s:conversation="
FulfillmentRequest" />
12 </extensionActivity>
13 </sequence>
14 .
15 .

```

Listing 7. Snippet of BPEL4SWS for Order Fulfiller(Lab)

This element enables grouping of interaction activities and thus enables defining a complex message exchange between two partners independent from WSDL. That means that <b4s:conversation> describes the communication between two partners without a dependency on WSDL. Lines 6-12 is a standard BPEL element <sequence> that enables sequential execution of business activities. Line 7 is a interaction activity <invoke> that enables communication with a partner Web Service(LAB EPR) referencing the portType and operation to be used. This <invoke> references a <partnerLink> which specifies the role the partner service and the process itself plays. In BPEL4SWS interaction activities have at least one variable, input and/or

output variable depending on their usage. In line 7 an attribute `modelReference` references an ontological representation (functional ontology) of the interaction activity (`<invoke>`). During invocation, functional ontologies are mediated upon to find equivalent invocation names in other healthcare standards.

```

1 <b4s:conversations>
2   <b4s:conversation b4s:name="OrderFulfiller"/>
3   <b4s:conversation b4s:name="OrderReceiver"/>
4 </b4s:conversations>
5
6 <sequence>
7   <b4s:receive name="Result" sa:modelReference="http://host/
      HL7v3FunctionalOntology#ResultComplete" partnerLink="
      OrderFulfiller" portType="spwsdl:OrderFulfillerPortType" operation=
      "Fulfiller_LabOrder" variable="LabResult" createInstance="yes"
      />
8
9   <extensionActivity>
10    <b4s:invoke name="ResultConfirmation" modelReference="http://host
      /HL7v3FunctionalOntology#ResultConfirmationResponse" b4s:
      inputVariable="ConfirmationResponse" b4s:outputVariable="
      ResultConfirmationResponse" b4s:conversation="
      FulfillmentRequest" />
11  </extensionActivity>
12 </sequence>
13 .
14 .

```

Listing 8. Snippet of BPEL4SWS for Order Receiver(Galway Hospital)

A detailed description of BPEL4SWS [9, 10] is outside the scope of this paper. Our focus is around PPEPR's integration requirements at the service and process levels and how BPEL4SWS resolves heterogeneity among various process models even if they implement a specific healthcare standard.

6. Lightweight PPEPR

The PPEPR ontologies and execution environment are based on the Web Service Modeling Ontology (WSMO). PPEPR uses the lightweight form of WSMO to meet healthcare enterprise integration needs. Below, we describe how PPEPR ontologies and the architecture are designed to be lightweight so that they are interoperable with other semantic languages and semantic Web service frameworks (SWS).

- WSMO applies a goal-based approach for the task of discovering Web services. The Goal (requirements) of a services requester and the capability (what it can offer) of the provider are semantically described to achieve (total or partial) automation in the complete Web service discovery process [19]. The PPEPR architecture is not goal based. End-points of EPR services are known to interacting services, thus service discovery is not the major focus of PPEPR.

- PPEPR uses the WSML language for describing domain-specific ontologies. PPEPR ontologies are lightweight in the sense that they use minimal common semantics (e.g. excluding 'axioms', 'relations', etc.), therefore PPEPR ontologies can be easily converted to other semantic languages, e.g. RDF/RDFS⁸, OWL⁹ without losing the semantics.
- PPEPR's ontologies and architecture are designed to align with WSMO-Lite [16] and SAWSDL [5] semantic Web service framework (SWS) for bottom-up healthcare service development and implementation. As SAWSDL is independent of any particular semantic technology PPEPR ontologies could be easily reused within the SAWSDL framework with minor or no modification.
- PPEPR's multi-party process models (or orchestrations) is based on BPEL4SWS where processes can be exposed both as semantic Web services and conventional Web services i.e., a valid BPEL4SWS document is also a valid BPEL document.

7. Related Works

COCOON [15] is a project aimed at setting up a set of regional semantics-based healthcare information infrastructure with the goal of reducing medical errors. In order to enable a seamless integration of eHealth services, Semantic Web Services technology is applied.

ARTEMIS [1, 15] aimed at developing semantic Web Services based Interoperability framework for the healthcare domain. Artemis has a peer-to-peer architecture in order to facilitate the discovery and consumption of healthcare Web services.

The major differences between COCOON, ARTEMIS and PPEPR are:

- COCOON and ARTEMIS are Web-scale projects. The major focus of PPEPR is to ease the integration burden of healthcare enterprises. Additionally, PPEPR's architecture is flexible enough to include Web-scale integration.
- ARTEMIS and COCOON employ primarily top-down approaches as far as semantics (ontology development) for service oriented architecture is concerned. PPEPR incorporates both top-down and bottom-up methodologies.

⁸<http://www.w3.org/RDF/>

⁹www.w3.org/2007/OWL/

- PPEPR defines the clear "separation of concerns" for services and healthcare process model.
- PPEPR ontologies can be easily used in other SWS frameworks such as SAWSDL.
- PPEPR ontologies are lightweight (using only a subset of WSML features). The major motivation behind this is to be interoperable with other standard semantic languages, e.g. RDF/RDFS, OWL.
- PPEPR applies a new mechanism to describe the communication between two partners without a dependency on WSDL. As described in Sections 3 and 5 BPEL4SWS introduces a new element, `<bs:conversation>`, which is not dependant on a `<partnerLink Type>` and as such is not dependant on WSDL. This element enables the grouping of interaction activities and thus enables defining a complex message exchange between two partners. For example, Artemis introduced business process template "BP template" to model business process at design time. BP template is dependent on WSDL (e.g. `<partnerLinkType>`) to describe a contract between two partners in terms of roles and corresponding WSDL portTypes. Also, for interaction activities "BP template" mainly relies on partnerLink and a WSDL operation. PPEPR clearly identifies the public and private behavior of interacting healthcare enterprises.

8. Conclusions

Any eHealth integration system, which connects healthcare enterprise applications must facilitate heterogeneous healthcare systems at all levels - data, services, processes, healthcare vendors, standards, legacy systems, and new information systems, all of which must interoperate to provide healthcare services.

In this paper, we describe the need for semantics in a service-oriented architecture (SOA) based healthcare integration system. We present an approach to ontologise interaction behavior of service-oriented enterprise healthcare that enables interaction between healthcare enterprises in presence of heterogeneous service and process definitions. The paper describes the latest results in the development of PPEPR, our integration system that connects enterprise healthcare applications at all levels (data, service, and process). PPEPR's architectural and ontological designs are domain based. These designs and ontologies include both standard based ontologies (functional and behavioral) and the definition of approaches used to develop them. PPEPR ontologies are lightweight to be interoperable with other semantic languages and semantic Web service (SWS) framework.

As next steps we plan to focus on optimizing ontologies. This will have the result of reducing the size of ontologies and mapping definitions. We plan to automate the grounding tasks (from XML/XMLSchema/WSDL to WSML and back) for both the HL7 versions (v2 and v3). We see this as PPEPR's core strength compared to syntactic integration solutions. In addition, we plan to incorporate example scenarios with more complex HL7 message exchange patterns within PPEPR.

9. Acknowledgement

This material is based upon works supported by the Science Foundation Ireland project Lion under Grant No.(SFI /02/CE1/I131) and by Enterprise Ireland under Project SAOR (CFTD 2005 INF 224).

References

- [1] V. Bicer, O. Kilic, A. Dogac, and G. B. Laleci. Archetype-Based Semantic Interoperability of Web Service Messages in the Health Care Domain. *Int'l Journal on Semantic Web and Information Systems*, 1(4):1–22, 2005.
- [2] M. L. Brodie, C. Bussler, J. D. Bruijn, T. Fahringer, D. Fensel, M. Hepp, H. Lausen, D. Roman, T. Strang, H. Werthner, and M. Zaremba. Semantically enabled service-oriented architectures: A manifesto and a paradigm shift in computer science. Technical Report TR20051226, DERI, 12 2005. Available at <http://www.deri.ie/fileadmin/documents/DERI-TR-2005-12-26.pdf>.
- [3] C. Bussler, D. Fensel, and A. Maedche. A Conceptual Architecture for Semantic Web enabled Web Services. *SIG-MOD Rec.*, 31(4):24–29, 12 2002.
- [4] E. Cimpian and A. Mocan. WSMX Process Mediation Based on Choreographies. In *1st International Workshop on Web Service Choreography and Orchestration for Business Process Management*, Nancy, France, 9 2005. IEEE Computer Society.
- [5] J. Farrell and H. Lausen. Semantic Annotations for WSDL and XML Schema, 8 2007. W3C Recommendation.
- [6] A. Filipowska, A. Haller, M. Kaczmarek, T. V. Lessen, J. Nitzsche, and B. Norton. Process Ontology Language and Operational Semantics for Semantic Business Processes. Available at <http://www.ip-super.org/res/Deliverables/D1.3.pdf>.
- [7] A. Honey, A. Dutta, J. Porrasmaa, J. Mykkanen, K. Connor, M. Kumar, and R. Stevens. Service Oriented Architecture and HL7 v3 Methodology. Available at http://www.hl7.org/v3ballot2008jan/html/infrastructure/soa4hl7/SOA4HL7_Methodology_final.pdf.
- [8] I. Iakovidis, A. Dogac, O. Purcarea, G. Comyn, and G. B. Laleci. Interoperability of eHealth Systems - Selection of Recent EU's Research Programme Developments. In *Proc.*

- of CeHR: International Conference 2007, eHealth: Combining Health Telematics, Telemedicine, Biomedical Engineering and Bioinformatics to the Edge, Regensburg, Germany, 12.
- [9] D. Karastoyanova, T. van Lessen, F. Leymann, Z. Ma, J. Nitzsche, B. Wetzstein, S. Bhiri, M. Hauswirth, and M. Zaremba. A reference architecture for semantic business process management systems. In *Semantic Web Technology in Business Information Systems (SWEBIS) Workshop*, 2008.
- [10] J. Nitzsche, T. van Lessen, D. Karastoyanova, and F. Leymann. Bpel for semantic web services (bpel4sws). In *OTM Workshops (1)*, pages 179–188, 2007.
- [11] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.
- [12] R. Ruggeri, M. de Graauw, C. Kaler, L. F. Cabrera, and M. Regio. HL7 Version 3 Standard: Transport Specification - Web Services Profile, Release 2. Available at <http://www.hl7.org/v3ballot/html/infrastructure/transport/transport-wsprofiles.htm>.
- [13] R. Sahay, W. Akhtar, and R. Fox. Ppepr: Plug and play electronic patient records. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing, the Semantic Web and Applications (SWA) track*, Fortaleza, Cear, Brazil, 3 2008. Available at <https://share.deri.ie/ppepr/paper/SCM-SAC2008.pdf>.
- [14] F. Scharffe and J. de Bruijn. A language to specify mappings between ontologies. In *Proceedings of The First International Conference on Signal-Image Technology and Internet-Based Systems (IEEE SITIS05)*, Yaound, Cameroon, 11 2005.
- [15] E. D. Valle, D. Cerizza, P. D. M. Veli, B. Yildirak, K. Gokce, B. Laleci, and H. Lausen. The Need for semantic Web Service in the eHealth, 6 2005. In W3C Workshop-SWSF, Innsbruck, Austria, Position paper.
- [16] T. Vitvar, J. Kopecky, J. Viskova, and D. Fensel. WSMO-Lite Annotations for Web Services. In *The 5th European Semantic Web Conference*, Tenerife, Spain, 3 2008. Available at <http://cms-wg.sti2.org/doc/ESWC2008-VitvarKVF.pdf>.
- [17] T. Vitvar, J. Kopecký, M. Zaremba, and D. Fensel. Wsmo-lite: Lightweight semantic descriptions for services on the web. In *ECOWS '07: Proceedings of the Fifth European Conference on Web Services*, pages 77–86, Washington, DC, USA, 2007. IEEE Computer Society.
- [18] T. Vitvar, A. Mocan, M. Kerrigan, M. Zaremba, M. Zaremba, M. Moran, E. Cimpian, T. Haselwanter, and D. Fensel. Semantically-enabled service oriented architecture: Concepts, technology and application. *Service Oriented Computing and Applications*, 5 2007.
- [19] M. Zaremba, T. Vitvar, and M. Moran. Towards optimized data fetching for service discovery. In *Proceedings of the Fifth IEEE European Conference on Web Services (ECOWS)*. IEEE, IEEE Computer Society, 11 2007.