1-1-2010

The health service bus: An architecture and case study in achieving interoperability in healthcare

Amanda Ryan
aryan@uow.edu.au

Peter W. Eklund
University of Wollongong, peklund@uow.edu.au

The health service bus: An architecture and case study in achieving interoperability in healthcare

Abstract

Interoperability in healthcare is a requirement for effective communication between entities, to ensure timely access to up to-date patient information and medical knowledge, and thus facilitate consistent patient care. An interoperability framework called the Health Service Bus (HSB), based on the Enterprise Service Bus (ESB) middleware software architecture is presented here as a solution to all three levels of interoperability as defined by the HL7 EHR Interoperability Work group in their definitive white paper "Coming to Terms". A prototype HSB system was implemented based on the Mule Open-Source ESB and is outlined and discussed, followed by a clinically-based example.

Keywords

health, service, bus, architecture, case, study, achieving, interoperability, healthcare

Disciplines

Business | Social and Behavioral Sciences

Publication Details

Ryan, A. & Eklund, P. W. (2010). The health service bus: An architecture and case study in achieving interoperability in healthcare. MEDINFO 2010 (pp. 922-926). Netherlands: IOS Press.

The Health Service Bus: An Architecture and Case Study in Achieving Interoperability in Healthcare

Amanda Ryan, [redacted]

S[redacted]lia

Abstract

Interoperability in healthcare is a requirement for effective communication between entities, to ensure timely access to up-to-date patient information and medical knowledge, and thus facilitate consistent patient care. An interoperability framework called the Health Service Bus (HSB), based on the Enterprise Service Bus (ESB) middleware software architecture is presented here as a solution to all three levels of interoperability as defined by the HL7 EHR Interoperability Work Group in their definitive white paper "Coming to Terms". A prototype HSB system was implemented based on the Mule Open-Source ESB and is outlined and discussed, followed by a clinically-based example.

Keywords:

Interoperability, Systems Integration, HL7, SNOMED CT.

Introduction

A main challenge in the field of health informatics is to enable shareable and computable information [1]. Taylor defines three "grand challenges" for health informatics: reading and writing patient records; creation of medical knowledge; and access to medical knowledge [2]. The key to all these challenges is interoperability, which is defined in the next section.

In [3], a conceptual framework for interoperability in healthcare was outlined based on the Java-based Jini Architecture. Following that work, due to some limitations of the Jini Architecture, an improved framework called the *Health Service Bus* based solely on Enterprise Service Bus technology but following the same principles was developed and is presented here.

Definition of Interoperability

The definition of interoperability in healthcare was the focus of research conducted by the HL7 Interoperability Work Group, a part of the Electronic Health Record Technical Committee. The result of this study was the observation that there are in fact three definitions of interoperability in the e-health industry [4]. In practice, these definitions correspond to three hierarchic levels of interoperability.

The first is *technical* interoperability, which involves communicating data between different applications and systems over a network. Messaging is the best solution to this type of communication problem [5], and is the approach taken here, as the Health Service Bus is built on a messaging core.

The second level, and the most difficult to achieve, is *semantic* interoperability. This pertains to messages not just being received, but that meaning can be transmitted from the sender to the receiver and be understood in the same way by both. Semantic interoperability is about the meaningful exchange of information in association with its context. It evolves beyond just communicating message structure into also communicating the intent or meaning of data, so that the information will be understood in precisely the same way by both the sender and recipient [6].

The third level is *process* interoperability, which refers to social or workflow engineering aspects of interoperability.

The ESB-based solution to interoperability in healthcare proposed addresses all three levels of interoperability as defined in [4], as will be shown.

Methods

The methods employed in this work follow the paradigm of *design science*. Design science is a method of information systems research which seeks to extend the boundaries of human and organisational capabilities by creating new and innovative artefacts [7]. Design science is technology-oriented research concerned with the creation of artificial artefacts which fall into four product-types: *constructs*, *models*, *methods* and *implementations* [8].

The work presented here as a solution to interoperability in healthcare is a model of an implementation, using distributed methods from the field of enterprise integration. The main method from that field employed here is that of the *Enterprise Service Bus*.

Enterprise Service Bus

Enterprise Service Bus (ESB) is a term coined at Stanford University to describe a specific middleware software architecture. An ESB provides a loosely-coupled, highly distributed approach to enterprise integration.

An ESB has a standards-based messaging engine, which is event-driven and provides foundational services for more complex software systems [9].

Standards-based integration is a fundamental concept of ESB and thus makes a fitting solution in the standards-filled realm

of health informatics. ESB solutions can also be implemented incrementally, so there is no need for downtime to completely swap over to a new architecture – it can be done one department at a time as it is deemed is necessary or convenient.

The Health Service Bus (HSB) presented here is essentially an ESB using health standards within its messaging formats.

Service Containers and Endpoints

In an ESB, all applications and services that are connected to the bus are considered abstract endpoints. The underlying implementations of these endpoints can be diverse, but the abstraction of treating them as the same provides a powerful paradigm for higher-level tools to assemble endpoints into process flows – a part of process interoperability.

A service container is a physical implementation of an endpoint and provides the service interface to the ESB. The traits of service containers lead to the distributed nature of ESB [9]. Figure 1 shows a generic endpoint.

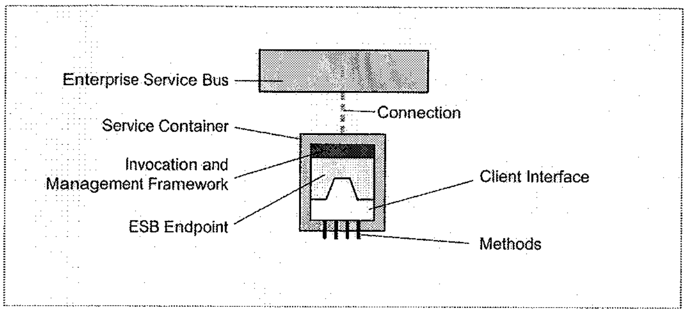

Figure 1 – A Generic ESB endpoint

Service containers should handle configuration, fault handling and management data – called the "Invocation and Management Framework". As Figure 1 shows, the endpoint connects to the client via an interface, which should be separate from the actual methods in the service connecting to the bus. This means that services can be changed over time, as long as their interface to the bus remains the same.

Mule Open-Source ESB

Mule, owned by MuleSoft, is the world's most widely used Open-Source ESB [10]. MuleSoft's definition of an ESB is as follows:

> *"An ESB functions as a transit layer for carrying information, providing connectivity to a wide range of heterogeneous technology assets. The bus provides a set of capabilities to enable integration and service-oriented architecture (SOA), including service creation and mediation, routing, data transformation, and management of messages between endpoints"* [10].

Like Jini, Mule is Java-based and provides a messaging framework which can use a variety of message formats. In Mule, application functionality is wrapped as a service, which includes a service component (business logic), routers (where to send the message), and configuration settings. This is consistent with the standard for ESB service containers and endpoints, and also provides process interoperability.

Results

HSB Prototype Implementation

A prototype HSB implementation was set up using the Mule Open-Source ESB, and making use of various services connected to the bus.

A client application for entering patient observations was developed for testing the prototype HSB. A SNOMED CT XML database, developed during a previous project was given a Web Services front-end and hooked into the HSB, allowing the Observations Application timely access to the SNOMED CT terminology.

Translation Services were also developed based on XSLT transforms to translate HL7 V3 messages to HL7 V2 and OpenEHR. The structure and content of these messages provides the level of semantic interoperability in the HSB. The actual XSLT transforms are discussed in [11].

The translation service was also set up with a Web Services front-end for connecting to the HSB. Figure 2 shows the components of the Mule HSB prototype.

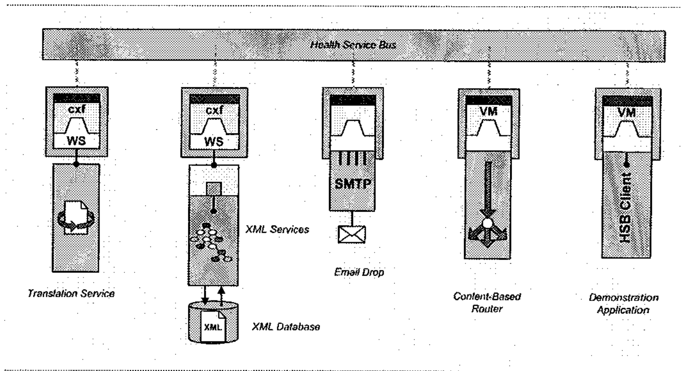

Figure 2 – Components of the Mule HSB prototype

As well as the services discussed, a content-based router is also included, and an email drop for reporting errors. The "HSB Client" shown in Figure 2 refers to the Observation Application, and the "XML Services" and "XML Database" combined make up the SNOMED CT XML Service.

Translation Service

Ontology mapping between different health standards (HL7 Version 2, HL7 Version 3 and OpenEHR) led to the development of XSLT stylesheets for translations between these standards in the instance of patient observations messages, as discussed in [11]. These XSLT stylesheets form the basis of the translation service connected to the HSB.

The functionality provided by the translation service allows for direct translation between HL7 V3 and HL7 V2, and from HL7 V3 and OpenEHR. Translation between HL7 V2 and OpenEHR may also be achieved within 2 steps – from V2 to V3 and then V3 to OpenEHR or vice versa.

SNOMED CT Terminology Service

The "XML Services" and "XML Database" in Figure 1 together refer to the SNOMED CT terminology service.

A subset of the SNOMED CT terminology was converted to XML for use as the basis of a terminology service which was connected to the HSB, based around the concept of *Vital Signs*. This encompasses Observations data, which is the example data used the HSB prototype.

SNOMED CT is distributed as data delimited into fields, with each record on a new line. To convert the SNOMED CT concepts to XML, each concept is converted to an XML node called "concept", with each field of the concept represented as one of its child nodes. For example, compare Table 1, which shows the concept "Sensation of blocked ears", to its XML representation, below.

Table 1 – Example SNOMED CT Concept

Concept ID	Status	Fully Specified Name	CTV3 ID	SNOMED ID	Primitive
103281005	0	Sensation of blocked ears (finding)	XU0sM	F-F5612	1

```
<concept>
   <conceptId>103281005</conceptId>
   <status>0</status>
   <fullySpecifiedName>
         Sensation of blocked ears (finding)
   </fullySpecifiedName>
   <CTV3Id>XU0sM</CTV3Id>
   <SnomedId>F-F5612</SnomedId>
   <isPrimitive>1</isPrimitive>
</concept>
```

The SNOMED CT relationships and descriptions pertaining to the concepts in the subset were also converted to XML in a similar manner. These XML records were then stored in a native-XML database, which provides a richer data model than a traditional database and makes the data self-describing, by way of the XML tag names [9]. The XML database can be queried over the bus by sending messages to its Web Services front-end.

Email Drop Service in the HSB

The purpose of the email-drop service is for automatically sending emails to a designated address when errors occur on the HSB. A service container is configured containing an endpoint which simply connects to an SMTP server and sends an email with a given subject and message to a defined address. The other services on the HSB are then configured to automatically use this service whenever an error occurs.

Content-Based Router

A *content-based router* examines the contents of a message and then routes it based on information contained in the message [5]. In the HSB, messages are routed based on who the messages are addressed to. HSB clients register their names with the router when they first connect to the bus. Messages between HSB clients are then sent through the router, which will pass them on to the appropriate party. The end-to-end example at the end of this section will demonstrate the router.

Patient Records in the HSB

Continuous patient records are stored in an XML database connected to the HSB. The format of the records is OpenEHR.

The format of most messages transmitted on the HSB is HL7 V3, so the translation service is used here to translate messages from the HSB into OpenEHR for storage in the database. The observation structures in HL7 V3 and OpenEHR are similar, making this translation simpler than expected. The "observation" sections of HL7 messages are translated to OpenEHR observations and then stored in the relevant patient's record.

Figure 3 shows the process of storing the observations in the patient's record. The setup of the EHR XML database is the same as the SNOMED CT database – the "XML Parser" in Figure 3 corresponds to "XML Services" in Figure 2, XML databases are shown in both Figures 2 and 3.

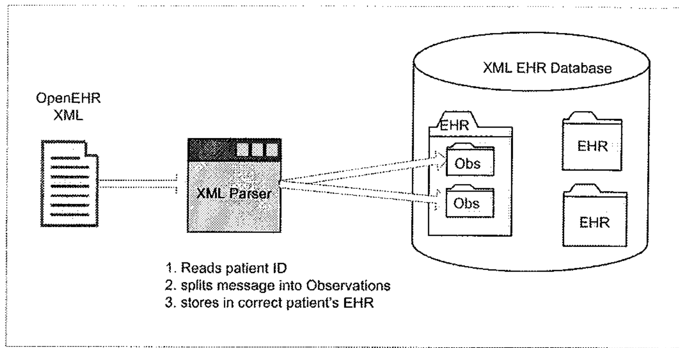

Figure 3 – Patient records stored in an XML database on the HSB. Observation Entries are stored in a patient's EHR.

Total Interoperability in the HSB

All three levels of interoperability (based on the definition in [4]) are achieved in the HSB.

Technical Interoperability

Technical interoperability is achieved in the HSB by the central HSB structure, specifically the messaging bus core. The messages sent "over the wire" are actually XML based on health standards, which brings us to semantic interoperability.

Semantic Interoperability

Semantic interoperability is achieved in the HSB by the content of the message sent within the framework and how they are processed. The messages are XML based on HL7 V3 models, which are themselves based on SNOMED CT constructs. The mapping from SNOMED CT to HL7 to obtain these models is covered in [12] and [13].

Process Interoperability

The level of process interoperability is achieved by the inherent nature of the HSB. Features such as intelligent routing and monitoring facilities which are part of ESBs contribute largely to process interoperability.

End-to-End Example

Figure 4 shows an end-to-end messaging example from the Mule HSB implementation. The HSB client wishes to send a message on the bus to another HSB client.

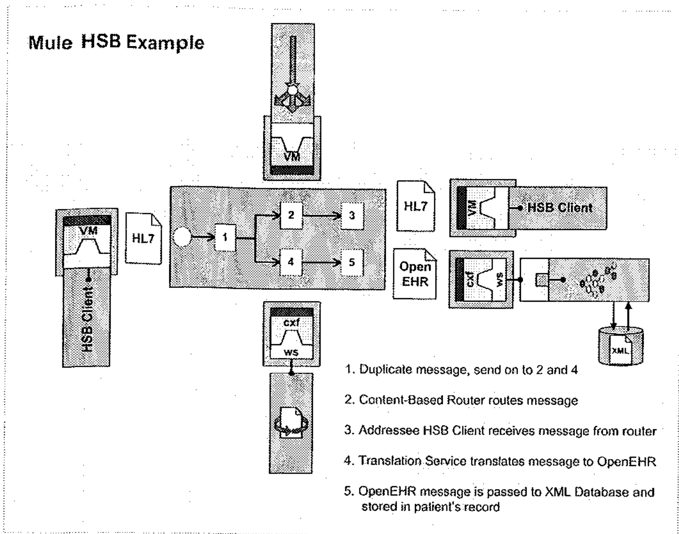

Figure 4 – End-to-end example of the Mule HSB implementation.

Step 1 is to duplicate the message and send it on to both the content-based router and the translation service. All observations messages sent from the HSB client Observations Applications are recorded in the OpenEHR database in the respective patient record.

Step 2 is the content-based router, which routes the message as described previously and sends it on to the receiver (Step 3).

Step 4 is the translation service, which translates the HL7 message into OpenEHR and passes it on to the EHR database (Step 5), where it is stored in the database as described previously.

Discussion

The prototype HSB demonstrates how ESB concepts can be applied to healthcare and is useful as a proof-of-concept. However, a large scale health information system is more complicated than this.

The HSB is scalable in that ESB is a highly distributed enterprise integration solution in that services that can work together are actually separated due to loose coupling principles and can be scaled independently from one another.

In terms of vertical scalability, a service container may manage multiple instances of a service; and in terms of horizontal scalability, several service containers may be distributed across multiple machines for the purpose of handling increased message volume [9].

Figure 5 shows an industry-level example between several healthcare facilities. Each entity has its own setup and can communicate within itself and also to all other entities in the network through the HSB. For example, the Aged Care Facility has three desktops with a packaged patient administration application installed on each. The three computers can communicate with each other, and the Aged Care Facility as a whole can also communicate with other entities in the care network.

Previous Work

Bicer et al's *Artemis Message Exchange Framework* (AMEF) aims to mediate between healthcare information systems with differing messaging standards using Web Services and OWL[14].

A prototype demonstration mapping HL7 V2 to HL7 V3 is shown by the authors in [14]. This involves ontology mapping of the HL7 V2 and V3 XML schemas to each other using a mapping tool they developed called *OWLmt.* Existing applications are then wrapped as Web services and the messages they exchange are mediated through OWLmt

The HSB is different AMEF in that it is a complete integration solution, encompassing message mediation, terminology services, patient records, management and monitoring and whatever else is required to be plugged into it; whereas AMEF concentrates solely on message mediation with some Web Services.

Conclusion

ESB is a powerful technology for standards-based integration, which provides am excellent solution for communication in healthcare. The HSB solution achieves all three levels of interoperability, as defined by the HL7 Interoperability Work Group – technical, semantic and process – thus providing an architecture for a complete interoperability solution.

The small proof-of-concept HSB using Mule ESB Open-Source software shows that the solution has potential to be adapted to a larger industry-based solution.

Acknowledgments

This work was supported by the Australian Research Council via its Linkage Grant Scheme LP0454061. We also acknowledge the support of Pen Computer Systems via this grant.

The *Sonic ESB Icon and Diagram Library* for *Microsoft Visio* was used for drawing Figures 1, 2, 4 and 5.

References

[1] OpenEHR Foundation. OpenEHR. http://www.openehr.org/. 2007. Last Accessed: September, 2009.

[2] Taylor P. From Patient Data to Medical Knowledge: the principles and practices of Health Informatics. BMJ Books. Blackwell Publishing Ltd. 2006.

[3] Ryan AJ and Eklund PW.' A Framework for Semantic Interoperability in Healthcare: A Service Oriented Architecture

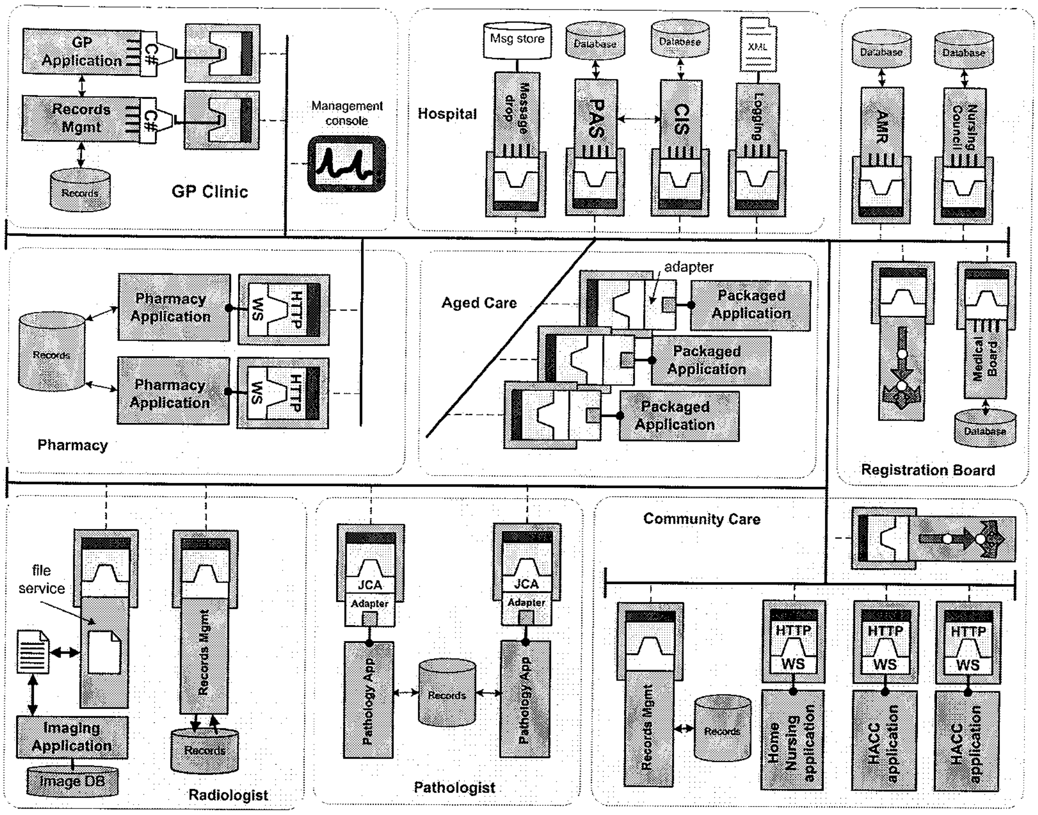

Figure 5 – Industry-based example HSB, covering eight different entities in a healthcare network.

based on Health Informatics Standards. Proceedings of MIE 2008, IOS Press, pp. 759-764, 2008.

[4] HL7 EHR Interoperability Work Group. Coming to Terms: Scoping Interoperability for Healthcare. http://www.hln.com/assets/pdf/Coming-to-Terms-February-2007.pdf. 2007. Last Accessed: September, 2009.

[5] Hohpe G and Woolf B. Enterprise Integration Patterns: Designing, Building and Deploying Messaging Systems. Addison-Wesley. 2004.

[6] Ryan A, Patrick J and Herkes R. Introduction of Enhancement Technologies into the Intensive Care Service, Royal Prince Alfred Hospital, Sydney. Health Information Management Journal. 37(1):40–45. 2008.

[7] Hevner AR, March ST, Park J and Ram S. Design Science in Information Systems Research. MIS Quarterly, 28:1, pp. 75–105. 2004.

[8] March ST and Smith GF. Design and Natural Science Research on Information Technology. Decision Support Systems. 15, pp. 251–266. 1995.

[9] Chappell DA. Enterprise Service Bus. O'Reilly. 2004.

[10] MuleSoft. Mule ESB. http://www.mulesoft.com/mule-esb-open-source-esb. Last Accessed: September, 2009.

[11] Ryan AJ and Eklund PW. Ontology Mapping between HL7 Versions 2 and 3 and OpenEHR for Observations Messages. Proceedings of the Health Informatics Conference, 2009, HISA pp. 40-46,, 2009.

[12] Ryan AJ. Towards Semantic Interoperability in Healthcare: Ontology Mapping from SNOMED CT to HL7 Version 3. Advances in Ontologies 2006, CRPIT 72. Australian Computer Society. 2006.

[13] Ryan AJ, Eklund PW and Esler B. Toward the Interoperability of HL7 v3 and SNOMED CT: a case study modeling mobile clinical treatment. MedInfo 2007. IOS Press. Pp. 626-630. 2007.

[14] Bicer V, Laceli GB, Dogac A and Yildiray K. Artemis Message Exchange Framework: Semantic Interoperability of exchanged messages in the Healthcare domain. Proceedings of the ACM SIGMOD International Conference on Management of Data, 2005.

Address for Correspondence

Amanda Ryan
9 Jermyn St
Eight Mile Plains QLD 4113
Australia
ajr883@uow.edu.au