User Centred Networked Health Care
A. Moen et al. (Eds.)
IOS Press, 2011

doi:10.3233/978-1-60750-806-9-265

Implementation of an Open Source Provider and Organization Registry Service

Markus BIRKLE[a,1], Benjamin SCHNEIDER[a], Tobias BECK[b], Thomas DEUSTER[b], Markus FISCHER[b], Florian FLATOW[b], Robert HEINRICH[b], Christian KAPP[b], Jasmin RIEMER[b], Michael SIMON[b], Björn BERGH[a]
[a] *Center for Information Technology and Medical Engineering, University Hospital Heidelberg, Speyerer Straße 4, 69115 Heidelberg, Germany*
[b] *Institute of Computer Science, University of Heidelberg, Im Neuenheimer Feld 326, 69120 Heidelberg, Germany*

Abstract. Healthcare Information Exchange Networks (HIEN) enables the exchange of medical information between different institutions. One of the biggest problems running a HIEN is the unique identification of the care providers. The provider and organisation registry service (PORS) has to provide a unique identifier for care providers. The concept and the implementation of PORS will be described in this article. Finally the PORS implementation will be compared with the Integrating the Healthcare Enterprise (IHE) profile for a Healthcare Provider Directory (HPD).

Keywords. provider and organization registry, eHealth, IHE, open source, regional health networks

1. Introduction

Healthcare Information Exchange Networks (HIEN) enables the exchange of medical information between different institutions. One of the biggest problems running a HIEN is the unique identification of the care providers like hospitals, health care professionals and so on. Heinze et. al. illustrated that problem on the example of a personal electronic health record (PEHR) which is implemented by the University Hospital Heidelberg (UHH) to improve the information exchange between UHH and other hospitals, primary care givers and the patient itself [1]. Today a unique identification for physicians, the so called "lebenslange Arztnummer" (lifelong physician number) and for organizations, the so called "Betriebsstättennummer" (permanent establishment number) are available in Germany. But they are only used for billing purposes and there is no central electronic registry for these values. The solution to address this issue is the development of a service called Provider and Organisation Registry Service (PORS).

Heinze et. al. developed a IHE-compliant concept how a PORS should be implemented [2]. This approach can solve the problem in German HIEN but it is also generic enough to be used in other countries because they implemented similar

[1] Corresponding author: Dipl.-Inform. Med. Markus Birkle, Center for Information Technology and Medical Engineering, University Hospital Heidelberg, Speyerer Straße 4, Heidelberg, Germany; E-mail: markus.birkle@med.uni-heidelberg.de

identification concepts for care providers. According to the concept of Heinze et. Al. a prototype was developed by students within the scope of a software master class [3]. The result successfully provided a proof of concept whilst it also showed that additional important functionality still needs to be added. In order to provide a full-feature version of the PORS that could also be deployed in a productive environment, a new project was initiated in cooperation with faculty of computer science at the University of Heidelberg. The knowledge and the experiences gathered within the development of the prototype lead to an extension of the original concept. Furthermore it was decided to release the resulting application as Open Source.

2010 the IHE (Integrating the Healthcare Enterprise[2]) published a draft of an IHE IT Infrastructure Technical Framework Supplement for public comments [4]. This Supplement describes an IHE Profile for a Healthcare Provider Directory (HPD). The HPD profile supports the management of healthcare provider public information in a directory structure and defines an interface to query the stored data. The HPD profile suggested the implementation of this service with a Lightweight Directory Access Protocol (LDAP) Server.

The extended concept and implementation of the PORS will be described in this article. Finally the result will be compared with the HPD LDAP Server implementation concept.

2. Method

Based on the concept of Heinze et. al. [2] and the already implemented prototype [3] the PORS was developed within an Information Systems Engineering internship of the faculty of computer science at the University of Heidelberg. The results will be released as an Open Source Software project via the Open eHealth Foundation under the Apache Software Licence 2.

In preparation to define the software architecture the HL7 v2 standard was analyzed in order to determine if the provided HL7 v2 messages will fit the PORS requirements. To adequately address the given requirements they have been enhanced and based on this the system and database architecture was developed.

The technical implementation is based on Java. For persistence purposes the Hibernate[3] framework is used. The data storage is realized with a PostgreSQL[4] database. The graphical user interface (GUI) for manual administration is based on Java Server Faces.

3. Result

3.1. Adapted HL7 v2 Messages

The messages already provided by the HL7 v2 standard have to be extended for the PORS. The functionality add, update and de- or reactivation of an entry is carried out by a HL7 v2 Master File Notification (MFN) message. Queries can be triggered by

[2] http://www.ihe.net/

[3] http://www.hibernate.org/

[4] http://www.postgresql.org/

Conformance Based Master File Queries (QBP) and the result is provided by a Segment Pattern Response (RSP). This messages are described below.

3.1.1. Master File Notification (MFN)

With a Master File Notification Message (MFN) shown in Figure 1 a PORS entry could be added, updated and de- or reactivated. To add an entry the MFE-1 segment should contain the value MAD. If an entry should be updated the MFE-1 segment should contain the value MUP. To deactivate an entry the MFE-1 segment should contain the value MDC. In case of entry deactivation the provision of a deactivation reason is mandatory. The reason can be provided in the STF-38 segment.

The success of this transaction is reported to the sending system by an HL7 v2 Master File Acknowledgement (MFK) message.

```
MSH|^~\&|SAP-ISH^sapr3t^002|UKHD^0999|PORS||201012151600||MFN^M02^MFN_M02|1234|P|2.5.1
MFI|PRO^Provider^UKHD0001||UPD|||AL
MFE|MAD|||0000001234|PL
STF|0000001234|179999900|Beckenbauer^Franz^JR^Dr|Internist|M|19501101|||999^UKHD^^
    1.2.276.0.76.3.1.78^Universitätsklinikum Heidelberg|
    00496221566736^00496221562000^franz@beckenbauer.de|Musterstraße 14^^
    Musterhausen^^12345^DE|20091211|||||||||||||||||||||||||
```

Figure 1. Exemplary Master File Notification Message

3.1.2. Conformance Based Master File Query (QBP)

With a Conformance Based Master File Query (QBP) shown in Figure 2 the PROS can be queried. Two query types are available. The Z80 PORS Query combines the given query criteria's by a logical "AND". The Z81 PORS Query combines the given query criteria's by a logical "OR". So any complex type of a query can be created.

The query result is submitted by a Segment Pattern Response (RSP) message.

```
MSH|^~\&|SAP-ISH^sapr3t^002|UKHD^0999|PORS||201012151600||QBP^Z80^QBP_Q11|1234|P|2.5.1
QPD|Z80^PORS Query^HL70471|1234|PRO^Provider^UKHD0001||||Beckenbauer
RCP|I||R
```

Figure 2. Exemplary Conformance Based Master File Query Message

3.1.3. Segment Pattern Response (RSP)

With a Segment Pattern Response (RSP) shown in Figure 3 the PORS answers a query message. The response contains the complete SFT Segment similar to the MFN Message.

```
MSH|^~\&|PORS||SAP-ISH^sapr3t^002|UKHD^0999|201012151600||RSP^Z80^QBP_Q11|1234|P|2.5.1
MSA|CA|1234
QAK|1234|OK|Z80^PORS Query^HL70471|1|1|0
QPD|Z80^PORS Query^HL70471|1234|PRO^Provider^UKHD0001||||Beckenbauer
MFE|MAD|||0000001234|PL
STF|0000001234|179999900|Beckenbauer^Franz^JR^Dr|Internist|M|19501101|||999^UKHD^^
    1.2.276.0.76.3.1.78^Universitätsklinikum Heidelberg|
    00496221566736^00496221562000^franz@beckenbauer.de|Musterstraße 14^^
    Musterhausen^^12345^DE|20091211|||||||||||||||||||||||||
```

Figure 3. Exemplary Segment Pattern Response Message

3.2. System Architecture

The PORS architecture is component based and service oriented. Subsequently you can find a short description of the main components.

3.2.1. Core / Controller

The core and the controller provide the main functionality of PORS. The controller takes over the internal message handling between the different components and handles the requests from the message interface and the graphical administration frontend. It translates each request into an internal task format. The core component is based on a multi-thread engine and enables a parallel processing of multiple tasks.

In the future a multi-processor parallelization approach could address problems that may occur during the processing of huge amount of simultaneous requests.

3.2.2. Communication Interface

PORS provides several interfaces. A HL7 message interface handles generic HL7 messages and a HTTP and SOAP interface handles appropriate encapsulated HL7 messages. The SOAP interface can also be used to connect another graphical user interface or external software component.

Last but not least a graphical administration interface was implemented. All PORS functionalities and administrative adjustments can be done over it.

3.2.3. Authentication and Administration Module

PORS implements a user and role based security management concept. Each PORS transaction (e.g. execution of a MFN message) requires a successful user login and associated role privileges that are required for the transaction.

There is also an LDAP interface implemented to handle user logins and role privileges via a global LDAP server.

3.2.4. Persistence and Database Module

The persistence layer and the search engine of PORS are implemented using the Hibernate framework. Data transfer objects are used for the internal data exchange.

The service uses a PostgreSQL database. PostgreSQL provides the possibility to enhance the functionality on the database layer by adding C program code. This is enhancement functionality used for the implementation of the Jaro-Winkler Algorithm [5] for the duplicate recognition functionality directly on the database.

3.3. Comparison of PORS and HPD

The comparison of the PORS implementation with IHE HPD shows some fundamental differences between the concepts. The IHE HPD profile expects the implementation of a LDAP infrastructure. The LDAP protocol provides out of the box well standardized interfaces to add, update, delete and query entries within an LDAP server. Duplicate recognition mechanisms are note supported from LDAP out of the box.

4. Discussion

The PORS implementation has shown to be a quite complex task as the software design has to be selected very carefully. Still first tests affirmed that the chosen software

architecture is adequate and the overall system is stable and capable of providing the required functionality.

Because communication within an IHE based HIEN infrastructure is based on HL7 messages the HPD solution implementing a LDAP server is not ideal. Also the missing duplicate recognition functionality is a problem for data quality. A HL7 to LDAP adapter and a duplicate recognition script have to be written. Another difficulty is the management of local id's because the LDAP schema has be updated by adding a new id. In big LDAP databases this may take a long time.

The Java based PORS implementation with an HL7 standard based message interface has solved all this problems. Within the PORS core engine we have the full functionalities of an object orientated programming language and problems like duplicate recognition can be solved directly.

Acknowledgements: PORS was implemented within a cooperation project between the Center for Information Technology and Medical Engineering of the Heidelberg University Hospital, the Software Engineering Group and the Database Systems Research Group at the University of Heidelberg. Within the master course Applied Computer Science the students could choose the internship Information Systems Engineering. During this internship, which duration is about one semester, the students have to handle a complete software development project. At the winter semester 2010 / 2011 the software to develop was the PORS. Special appreciation is given to the professors Prof. Dr. Barbara Peach and Prof. Dr. Michael Gertz who headed this course. Many thanks also to Dipl.-Math. Florian Flatow and Dipl.-Inf. Robert Heinrich for the perfect supervision of the project. And last but not least a special thanks to the student's Tobias Beck, Thomas Deuster, Markus Fischer, Christian Kapp, Jasmin Riemer, Michael Simon for there extensive contribution to the project.

References

[1] Heinze, O. Brandner, A. Bergh, B. Establishing a Personal Electronic Health Record in the Rhine-Neckar Region. In Proceedings of MIE 2009 – The XXIInd International Congress of the European Federation for Medical Informatics, *Stud Health Technol Inform* (2009), 150-119.

[2] Heinze, O. Ihls, A. Bergh, B. Development of an Open Source Provider and Organization Registry Service for Regional Health Networks, *Third International Conference on Health Informatics* (HealthInf 2010), Spain, 2010.

[3] Doods, J. Porzelt, J. Stricker, L. Berz, C. Chruscz, K. Duda, et.al., *Technische Umsetzung eines regionalen Provider and Organization* Registry Service, GMDS Jahrestagung, Germany, 2010.

[4] Kande, S. Jain, N. Briks-Fader, T. Witting, K. *IHE IT Infrastructure Technical Framework Supplement, Healthcare Provider Directory (HPD*), Draft for Public Comment, IHE International, 2010

[5] Winkler, W. W. Overview of Record Linkage and Current Research Directions, *Research Report Series*, Statistics 2006-2, 2006.