A Hierarchical Event-based Architecture for the Notification of Medical Document Availability

Christian Esposito and Mario Ciampi

Institute for High Performance Computing and Networking (ICAR)
National Research Council of Italy (CNR), Napoli 80131 - Italy
christian.esposito, mario.ciampi@na.icar.cnr.it

Abstract. A patient typically receives secondary care from a hospital-based physician different than his/her general practitioner. The communication between the two of them is often tough and with deficits, which may negatively affect the continuity of care. In this paper, we present a system based on publish/subscribe services for the fast and effective notification of medical data between different healthcare providers. We introduce the concept of topic hierarchies so as to notify the availability of a collection of documents, somehow related. Moreover, we also assess the achievable performance of the implementation of such a system.

Keywords: Health Information Systems; Event Notification; Content Filtering

1 Introduction

In healthcare, general practitioners may refer their patients to secondary care physicians for clinical analysis, specialist referrals and/or surgeries. It is not rare that such hospital-based physicians reside quite far away from the general practitioners. In general, there is a communication issue that negatively affect the exchange of patient information between them. In fact, the patients are generally responsible of picking up paper-based clinical documents (such as medical reports or discharge letters) from where they have been produced and bringing them to their general practitioners. Recent studies, such as [1], have proved that such a paper-based communication is inefficient, error prone and too slow in many cases. Moreover, they also provided evidences that computer-based communication has positive effects, by improving the health care efficiency and safety, and reducing the overall costs [2]. Therefore, it is an increasing practice to use proper tools to support a computer-based communication in healthcare, such as the adoption of Electronic Health Records (EHR) [2]. An EHR is a collection of digital healthcare information regarding a patient, generated by his/her encounters within the healthcare, *e.g.*, hospital-based physicians. However, typical EHR systems are based only on pull-based communication strategies, *i.e.*, the information is delivered to the interested user only after he/she has made a direct request for it. The main flaw of this communication mode is that the user has to know what to request and if the interested information has been produced.

In this paper, we illustrate a solution for communicating medical data availability by means of publish/subscribe services [3], where notifications communicates the availability of a collection of documents, somehow related. In this sense, our solution overcomes the mentioned limitation of typical EHR systems by allowing a push-based communication mode, *i.e.*, the availability of documents of interest is immediately notified to users, so that they can make a direct request to access them. The proposed notification architecture has been implemented and tested so as to evaluate the achievable performance.

2 Background

A typical medical supply chain is mainly composed by general practitioners, which represent the first point of contact between the patients and the healthcare system, and have a complete view of the clinical history of their patient. The second main participants are healthcare providers, such as hospitals, laboratories or other medical institutions, which provide secondary care with additional health services to complement the ones offered by general practitioners. In fact, during their life, patients may receive treatments and/or perform analysis in such secondary care providers, either due to prescriptions from the general practitioners (*e.g.*, specialized medical tests such as Magnetic Resonance Imaging (MRI), or routine tests such as X-rays, cholesterol checks, blood-sugar checks) or due to patient initiatives (*e.g.*, plastic surgeries or treatments for diseases occurring during the holidays). General practitioners need to promptly receive outcomes of the secondary care so as to better follow up the patient clinical story.

A number of approaches are currently emerging to overcome the current fragmented view of hospital information systems that do not share data among themselves or even with general practitioners. An approach is the one proposed in [4], which is built on top of a middleware using the Tuple Space (TS), *i.e.*, an implementation of the associative memory paradigm for parallel/distributed computing that provides a repository of tuples presented as local even if tuples may reside to different distributed storages. In [4] medical data is seen as tuples stored and retrieved by means of TS, and flexibility is provided by offering proper adapters between TS and clients by means of WS APIs so as to bind asynchronous WS calls to operations on the TS. Anyway, the most considered approach to make HIS interoperable each other makes use of Web Services (WS), such as in the Italian HIS called InFSE[1]. The core business of these solutions is to store medical documents and to make them available to geographically-sparse interested users. This is able to partially satisfy the communication needs among general practitioners and physicians, leaving unresolved the issue of how notifying to users that a certain medical document of interest is available and can be retrieved. In this work, we address such a problem by means of a proper notification infrastructure. Due to the needs for an asynchronous, scalable and flexible communication, our infrastructure is based on the publish/subscribe paradigm, which has been widely adopted to federate and integrate a large number of

[1] http://ehealth.icar.cnr.it/

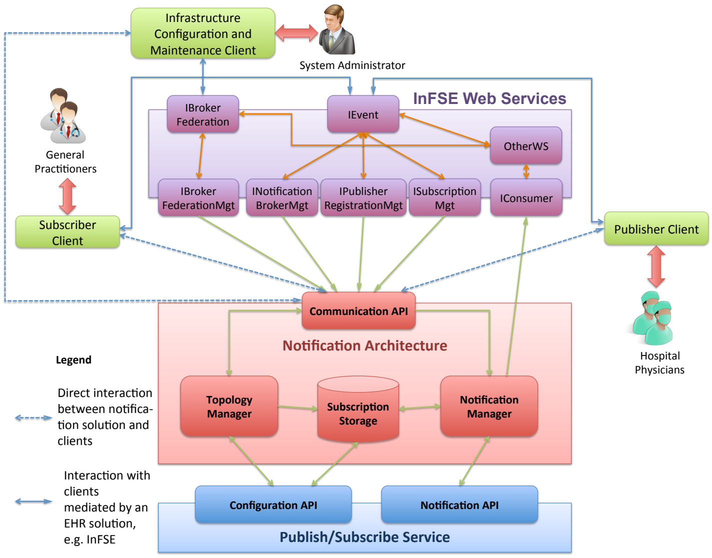

Fig. 1. Architecture overview of the proposed notification system.

systems [5]. The final goal is not to present an other novel interoperability infrastructure that competes with and substitute the ones previously mentioned. On the contrary, the scope is to present a system that can be put beside them so as to improve their efficiency and use.

3 Notification Architecture for Medical Data

3.1 Architectural Overview

Fig. 1 depicts an overview of the proposed architecture, based on a three-tier organization. At the lowest tier there is the publish/subscribe service, which can be of any given type. Due to the high support to scalability and efficiency, we have used in our system a product compliant to the JMS standard, specifically Apache Active MQ[2]. The functionalities offered at this tier are grouped as *Configuration API* and *Notification API*. The first group allows the user to define the topology of the infrastructure and the subscriptions of interest; while, the second one manages the production and reception of notifications. Built on top of the publish/subscribe service, there is our notification architecture, composed of three key components. The *Topology Manager* properly drives the configuration functionalities of the lower tier based on user inputs. The *Subscription Storage* keeps all the subscriptions; while, the *Notification Manager* disseminates the received notifications, based on the current subscriptions so as to delivery only data of interest. The top of this tier has a set of WS, composing the InFSE

[2] http://activemq.apache.org

infrastructure, with five different interfaces [6]: (*i*) *IBrokerFederationMgt* defines the topology of the system; (*ii*) *INotificationBrokerMgt* defines topics and retrieves events in a pull-based manner; (*iii*) *IPublisherRegistrationMgt* advertises the events that will be published; (*iv*) *ISubscriptionMgt* specifies subscriptions; and (*v*) *IConsume* consumes notification in a push-based manner. The other two WS in the figure are used as access points for the users. However, a user is free to use our notification solution directly, even without these web services (as depicted in figure). We have not fixed a certain format for the exchanged data, *i.e.*, internal structure of exchanged events, in terms of simple and/or complex data types. Thus, at the application level, it is possible to express any format of interest, *e.g.*, based on HL7 CDA specification.

It is worth noting that underlying our system there is an event broker, which can be interconnected according to a two-levels topology. Given a certain region, such as a country, we can have a federation of brokers belonging to different regions (level one), but also a federation of brokers within the same region (level two). In case of the presence of the second level federation, an hybrid peer-to-peer organization is realized: one of the brokers is elected as a gateway between the federation of first level and the one of second level. Such a gateway can be indicated by the user when the federation is defined, or it is possible to adopt a distributed protocol.

3.2 Enhanced Publish/Subscribe Service

JMS describes a topic-based publish/subscribe service [3], where publishers tag outgoing notifications with a topic string, while subscribers use string matching to detect their events of interest. Therefore, in our architecture a client can subscribe to one or more topics, intended as the possible types of medical data that the system can disseminate, such as results of blood or urine analysis, discharge letters or radiological reports. However, such a coarse-grain granularity of subscripting is not adequate in the domain of healthcare, since a general practitioner would not be interested to be notified whenever, for instance, a new blood analysis report is generated within the whole region covered by our system. In fact, it is more reasonable that he/she would be only interested in receiving notifications of blood analysis reports related to one of his/her patients. Therefore, to improve the effectiveness of the adopted subscription scheme, we have introduced a content-based filtering within the notification manager. Such a filter is applied on the identity of the patient whose the notification is related. Since several privacy regulations do not allow the presence of personal data and medical data within the same electronic artifact, notifications contain a unique anonymous identifier of the patient, so as to protect his/her privacy.

General practitioners are often not interested to the notification of single documents, but of a certain set of documents, somehow correlated. Therefore, a topic-based subscription, even with content-based filtering, is not adequate, since it can generate a number of unwanted notifications. For this reason, we have introduced the concept of hierarchy of topics. Specifically, a hierarchy is a set of topics linked among each others by proper associations, that can be of two types: AND, when notifications of two topics have to be provided to the user at the same time only if the difference between their respective production times does not exceed a certain threshold, and OR, when two topics are alternative. These boolean operators establish a hierarchical order on the topics describing a diagnosis plan or a clinical workflow. Content-based filtering can be applied to single topics, but also to hierarchies. For a concrete example, to diagnose if a patient has a celiac disease, it is needed to examinate a pair of specific antibodies: the Anti-Gliadin Antibodies (AGA) and Anti-Endomysial Antibodies (AEA). In addition, a test of Anti-Transglutaminase Antibodies (ATA) is equally precise and reliable than the one for AEA, but is much more simple and economical to produce. Therefore, the consumer will make a submission to the following hierarchy: {AGA Topic} AND ({AEA Topic} OR {ATA Topic}).

3.3 Integration with Existing HIS

As above mentioned, our notification solution can be used directly by users with a proper client application, but it can be also integrated within the context of an existing HIS, such as InFSE, or even advanced medical platforms as [7, 8]. InFSE allows creating clinical documents and storing them within the context of an EHR for a given patient. In addition, it also allows the retrieval of such documents and provides an abstraction of a single EHR even if the documents composing it are sparse in different regions integrated by InFSE. We have integrated our notification solution with the component in InFSE that is responsible for the creation of clinical documents, named as IDocument. As soon as IDocument is successfully invoked by a user so as to generate and store a new clinical document, the publish operation on IEvent is automatically called so as to notify the availability of the recently produced document. The notification is constructed by extracting the description of the document from its metadata, according to the HL7 Clinical Document Architecture (CDA)[3], so as to request and obtain the complete document from IDocument.

4 Experimental Campaign

We have conducted a series of experiments in order to evaluate the performance of our notification architecture. We have used InFSE to generate new documents, which are immediately notified through our notification solution. In particular, we have focused on the push-based event notification, and defined three different test scenarios: (*i*) Local Scenario, where our notification architecture, publisher and subscriber applications are running on the same machine; (*ii*) Distributed Scenario A, where our notification architecture and the two applications are running on two distinct machines; and (*iii*) Distributed Scenario B, where each machine runs an instance of our notification architecture and a publisher or subscriber application and the interconnection between two instances of our

[3] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44429

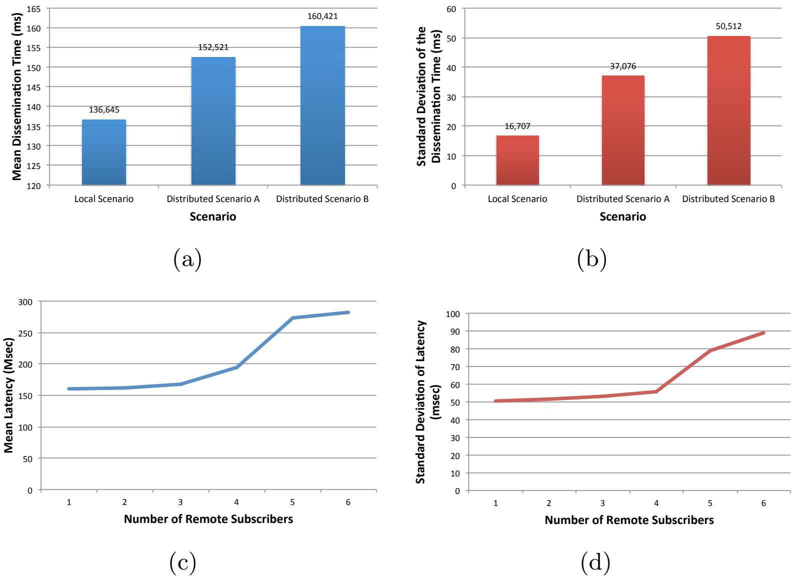

Fig. 2. Average and standard deviation of the delivery time in our test scenarios.

notification architecture is realized thanks to a network of two brokers of Apache MQ, configured by two Administrator applications (one per each machine) by using services offered by the IBrokerFederationMgt interface. In our distributed scenarios, we have used nodes taken from the *PlanetLab* [9], which is a well-known real wide-area testbed. In particular, due to the Italian scale of our project of reference, *i.e.* InFSE, we have considered the Italian nodes of PlanetLab. We started with a node in Torino and another in Trento (when we needed only two nodes in our Distributed Scenario B). Then, we have added more remote nodes when repeating the second distributed scenario. During a single experiment our publisher application produces 100 notifications of 300 Bytes.

In Fig. 2(a), we can see the delivery time in different test scenarios, where only one topic has been created and no content filter has been defined. In particular, for the local scenario, we have that the mean delivery time is equal to 136,645 milliseconds. We have performed such a test with varying the number of topics and hierarchies created by the publisher and subscribed by the subscriber, but we have not experienced a considerable degradation in performance (*i.e.*, the degradation with an additional topic is equal in average to 4,831 milliseconds). We have also replicated such experiments with the subscriber application defining a content-based filter on the value assumed by the patient id. The overhead registered for performing the filtering is about 1,737 milliseconds. When we have passed to more distributed scenarios, we have noticed an increase in the experienced dissemination time due to exchanging SOAP messages among the distributed applications. Specifically, such an increase is respectively 15,876 milliseconds for the first distributed scenario and 23,776 milliseconds for the second one with respect to the local scenario. Also the standard deviation of the experienced dissemination time follows such a tendency: it is low for the local scenario, *i.e.*, 16,707, and it increases when we consider a more distributed one.

We have repeated the last experiment by increasing the number of remote subscribers, which are distributed along the Italian peninsula, with a topology that is representative of a first deployment of InFSE. The network latency of the connections among the chosen nodes spans from an average of 23,4 milliseconds between Torino and Trento to 33,4 milliseconds from Catania to Trento. For low numbers of remote subscribers, *i.e.*, until 3 subscribers, the measured latency is about constant; however, when the number of remote subscribers is increased over such a threshold of 3, we experience a growth both in average and in the standard deviation of the measured latency. For the mean latency, the increase is about the 33%; while for the standard deviation, the increase is about 20%.

5 Related Work

Publish/subscribe services have been used in ubiquitous systems for continuously monitoring patients, with an event representing a measure of a certain vital sign. The intent is twofold. Since the delivery of every measure is inefficient, only those measures that are higher than a given threshold, that indicates a critical situation for the patient, are sent thanks to proper content-based subscriptions [10]. In addition, since more than one medical device is used in a typical assistive environment, each with a proper communication protocol, a publish/subscribe service is used to offer medical device interoperability thanks to its decoupling properties [11]. These solutions have become mature to be commercially marketed, such as QNX Medical Demo[4]. Our work differs from those since our scope is not to raise alerts or to identify meaningful threats, but to notify the final user of the availability of medical documents so as to have a timely retrieval. A work more similar to ours is the one presented in [12] where a publish/subscribe service is used as a mediator between inter-institutional HIS. However, this work is more focused on security aspects. On the contrary, we have focused on optimizing the effectiveness of such a communication by introducing topic hierarchies and content filters, and we consider the investigation of non-functional properties, such as reliability and security, as a future work. Another similar work is the one in [13], which specifies the use of SMTP and related standards for sending notifications of clinical documents. Our work is not limited to email notifications, and provides advanced functionalities, such as hierarchies of topics.

6 Conclusions

We have presented an architecture for notifying medical data availability, which meets the patient needs of an improved communication within the healthcare

[4] http://www.qnx.com/

by allowing the timely consultation of the results of treatments and/or tests made at hospitals. Our future work is to investigate on the Quality-of-Service guarantees of our solution, and which additional mechanisms may be needed.

Acknowledgements

We thank the partial support of the Italian project "ASK-Health" - Advanced System for the Interpretations and Sharing of Knowledge in Health Care.

References

1. S. Kripalani et al., "Deficits in communication and information transfer between hospital-based and primary care physicians," *The Journal of American Medical Association (JAMA)*, vol. 297, no. 8, pp. 831–841, 2007.
2. R. Hillestad, J. Bigelow, A. Bower, F. Girosi, R. Meili, R. Scoville, and R. Taylor, "Can Electronic Medical Record Systems Transform Health Care? Potential Health Benefits, Savings, And Costs," *Health Affairs*, vol. 24, no. 5, pp. 1103–1117, 2005.
3. P.Th. Eugster et al., "The many Faces of Publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, June 2003.
4. R. Krummenacher et al., "Enabling the European Patient Summary through tuple spaces," *Computer Methods and Programs in Biomedicine*, vol. 95, no. 2, pp. S33–S43, August 2009.
5. L. Fiege et al., "Publish-Subscribe Grows Up: Support for Management, Visibility Control, and Heterogeneity," *IEEE Internet Computing*, vol. 10, pp. 48–55, 2006.
6. M. Ciampi et al., "On Federating Health Information Systems," *Proceedings of the International Conference in Green and Ubiquitous Technology*, July 2012.
7. A. Coronato and G. De Pietro and I. Marra, "An open-source software architecture for immersive medical imaging," *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pp. 166–170, July 2006.
8. L. Gallo and G. De Pietro and A. Coronato and I. Marra, "Toward a natural interface to virtual medical imaging environments," *Proceedings of the working conference on Advanced visual interfaces*, pp. 429–432, May 2008.
9. A. Bavier, M. Bowman, D. Culler, B. Chun, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating System Support for Planetary-Scale Network Services," *Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI 04)*, pp. 253–266, March 2004.
10. D. Guerra and U. Gawlick and P. Bizarro, "An integrated data management approach to manage health care data," *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pp. 40:1–40:2, 2009.
11. A. King et al., "A publish-subscribe architecture and component-based programming model for medical device interoperability," *ACM SIGBED Review*, vol. 6, no. 2, pp. 7:1–7:10, July 2009.
12. C.P. Neumann et al., "A mediated publish-subscribe system for inter-institutional process support in healthcare," *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pp. 14:1–14:4, 2009.
13. Integrating the Healthcare Enterprise (IHE), "Notification of Document Availability (NAV)," *IHE IT Infrastructure Technical Framework Supplement*, August 2010.