Informatics for Health: Connected Citizen-Led Wellness and Population Health
R. Randell et al. (Eds.)

doi:10.3233/978-1-61499-753-5-33

Design, Implementation and Operation of a Reading Center Platform for Clinical Studies

Lucien CLIN[a,1], Martin A. LEITRITZ[b], Johannes DIETTER[c], Marek DYNOWSKI[d], Oliver BURGERT[a], Marius UEFFING[c] and Christian THIES[a]
[a] *School of Informatics, Reutlingen University, Germany*
[b] *University Eye-Hospital, University of Tübingen, Germany*
[c] *Institute for Ophthalmic Research University of Tübingen, Germany*
[d] *Now at: Cancer Research UK Manchester Institute, The University of Manchester, UK*

Abstract. Clinical reading centers provide expertise for consistent, centralized analysis of medical data gathered in a distributed context. Accordingly, appropriate software solutions are required for the involved communication and data management processes. In this work, an analysis of general requirements and essential architectural and software design considerations for reading center information systems is provided. The identified patterns have been applied to the implementation of the reading center platform which is currently operated at the Center of Ophthalmology of the University Hospital of Tübingen.

Keywords. Medical Information Systems, Software Platform, Clinical Studies, Telemedicine, Distributed Health Care, Software Engineering

1. Introduction

Reading centers bundle the knowledge of medical experts in a special field to analyse data and provide diagnoses. In the context of clinical studies, reading centers have the potential to ensure consistent high quality by reducing variation and bias via centralized analysis of the data according to fixed SOP [1]. However, this advantage of higher data quality comes at the cost of increased complexity and calls for robust software solutions to address all involved aspects of secure communication, data management, process configuration, visualization, as well as maintenance issues arising over the long-term operation of such a center.

In this work, an analysis of general requirements and ensuing essential architectural and software design considerations is provided, based on the implementation of the reading center platform operated at the Center of Ophthalmology of the University Hospital of Tübingen. The system concept follows the roles and processes defined in IHE Eye Care[2]. But it has been adapted to the needs of the project, e.g. there is no DICOM node available on the data provider side, and data formats were pre-defined by the application field. For that reason, it was considered expedient to

[1] Corresponding author: lucien.clin@reutlingen-university.de
[2] https://www.ihe.net/Eye_Care

develop an open platform that offers flexible modelling of varying distributed reading processes and efficient system interoperability for current standards as well as proprietary components.

2. Method

2.1. Reading Process and System Requirements

The general reading process has been described by Lotz *et al* in [2] as: Data from an examination performed on a *patient* at a *local* site is submitted to a *reading* center, where it is read by a user fulfilling the role of *reader* according to a pre-defined protocol (e.g. study SOP). In some contexts, the data may have to be read independently by two readers ("four eye principle"), with approval by a *senior reader* required in the case of discrepancies. The validated data forms the basis of a *report* returned to the local site.

The use case we are addressing here is an ophthalmology reading center operated at the Center for Ophthalmology of the University Hospital of Tübingen. The data consists of retinal fundus images together with, optionally, visual results or other study-relevant findings (blood pressure, lab results) or background data (questionnaires). Only pseudonymized data is handled within the reading center; all personal identification data remain with the originating local site.

Before entering the reading process, the examination data may be subjected to automated processing. In the present case, for instance, one specific feature of the reading center is the calculation of the artery-to-vein ratio using automated vessel recognition software [3]. For this purpose, the system is configured to submit image-processing jobs to a HPC cluster[3]. From this exemplary process, the following general requirements may be identified for a reading center information system:

1. Integration of data of external origin into a --- potentially pseudonymous --- electronic health record (EHR)
2. Communication with data processing modules (e.g. HPC Cluster) for "further added value"
3. Role- and SOP-based data access, visualization and examination work flow

2.2. System Architecture and Implementation

In the spirit of the "Platform as a paradigm" approach advocated by openEHR [4], in particular the principle that a "platform is a process, not a product", design decisions were primarily grounded in sound software engineering practices, with long-term maintainability in mind. The platform thus follows the service-oriented architecture (SOA) paradigm. In view of the inevitable need for change that arises during the operation use of such a system, a layered design based on established software design patterns has been adopted for loose coupling among individual components and thus minimize implementation effort for modifications. In particular, a metadata approach has been followed for all aspects pertaining to data modelling, from automatic

[3] https://www.bwhpc-c5.de

generation of code for data classes to their flexible mapping onto persistence mechanisms.

The data integration platform (server) has been implemented as a Java web application run on Apache Tomcat[4]; the graphical user interface (GUI) is a rich client written in C++ with the Qt framework[5]. In the following sections, details on the implementation and interplay of the various components are given.

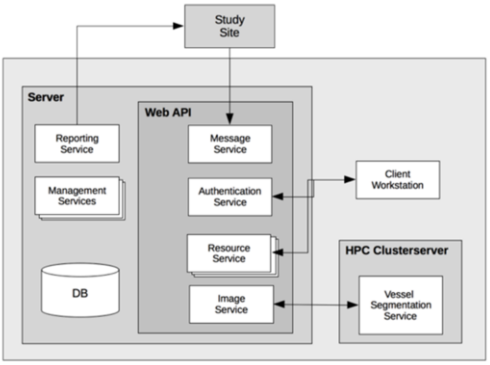

Figure 1. System architecture overview.

2.2.1. Data Exchange

The exchange of data with external sites (i.e. incoming new data to be analyzed, outgoing reports), as well as between the reading center server and GUI-clients, is based on HTTP with XML messages. It is therefore compliant to 'FHIR Messaging' in terms of protocol and serialization format, but ad hoc XML schemata defined for the use case at hand are used instead of adhering to pre-defined FHIR resources. This messaging approach has been chosen instead of a RESTful design because it better suited the event-based processes: external messages with new data initiate the reading process (download of specified images, submission of images to segmentation, notification of readers, etc.); incoming assessments/analyses from readers trigger the comparison for discrepancies or report generation. For secure communication, SSL/TLS is used with mutual server-client authentication based on X.509 certificates.

The Request Mapper Pattern [5] is applied here in order to decouple message formats from internal resource representations and thus allow for their independent evolution. It is implemented using XSLT to transform incoming messages, after which they are unmarshalled into Data Transfer Objects (DTO) [5] using an XML serialization framework[6]. Dealing with a new message format can thus be achieved via mere insertion of an appropriate XSLT script. By this approach interoperability standards, such as the Clinical Document Architecture, are realized as specific configurations for XML serialization.

---

[4] http://tomcat.apache.org

[5] https://www.qt.io

[6] http://simple.sourceforge.net

2.2.2. Data Modelling and Persistence

The chosen persistence mechanism is a hybrid of relational database (PostgreSQL[7]) and file system storage: A relational schema consisting of the core entities which are to be dealt with in the reading center forms the data model's foundation. All further data is stored as XML-, image- or other files in the file system, references to which are maintained in the relational database. Study-specific data is thus stored in a semi-structured way.

Loose coupling between the server application and the data stores is achieved using a custom persistence framework based on the design presented in [6]. Metadata is here used to generate the code of domain classes and to provide their mapping onto a database.

2.2.3. User Interface

The GUI-client is based on the Application Controller [7] and Model-View-Presenter patterns for modularity and extensibility. The functionality for the aforementioned reader and senior reader modes is implemented using role-specific controller/view components to provide task-specific work flow and visualization: The basic mode guides the reader through the analysis of the patient data (e.g. retinal photographies and other data) according to a study-specific examination catalogue. In the other mode, the senior reader is presented with the patient data and a comparison of the diverging findings with discrepancies highlighted, the successive resolution of which provides an approved report.

3. Results

The presented reading center platform is in operation at the Center for Ophthalmology of the University of Tübingen. After the initial development phase, two small studies with data volumes of 477 and 805 individual examinations (2011 and 3076 images, respectively) were performed on the system by two readers and one senior reader over a period of 7 months; these served as use cases to make final adjustments. At present, the reading center serves to provide examination reports for the ophthalmic data of the "German National Cohort" (NaKo7), a long-scale cohort study. In this context, retinal images of approximately 40.000 patients will be examined over the next years.

4. Discussion

The overall system design and implementation has been validated by successful operation in three studies so far. The applied design principles and technologies (HTTP/XML messaging, O/R-mapping, etc.) are well-established in enterprise application development, and can therefore be directly applied to any clinical information system, irrespective of the domain.

However, the platform is still at a prototypical stage and somewhat tightly coupled to the use cases which guided its implementation. To begin with, many of the processes

[7] https://www.postgresql.org

are hard-coded, in particular in the client GUI. Also, a high level of expert knowledge of the system's inner workings is still required for maintenance operations (user/role management, etc), which need to be performed manually, i.e. at 'system level', for lack of graphical tools. Although, of course, a certain process specificity and administration expertise is inevitable in any information system, we plan to address these issues in a further development iteration. In particular, we intend to extend the metadata approach to dynamic configuration of the client GUI, in particular as concerns study-specific setup of view components and workflow adaptation. In addition, further planned features include a query module (with role-specific access) to access the semi-structured data stored as XML documents; corresponding query forms will here be generated from the modelling metadata. For this purpose, the developed persistence framework is currently being generalized to provide a single API for dealing with further persistence mechanisms, especially the XML database BaseX[8] in addition to relational databases, together with use-case-specific resource representation via configurable serialization mappings. As a general objective, we intend to generalize the core components of this application to provide a basic toolkit for construction of similar platforms.

5. Conclusion

This work analyses essential architectural features and identifies software design patterns for reading center platforms. The proposed design has been implemented in an ophthalmic reading center software system. Since the applied principles and techniques are well-established in enterprise application development, the approach is straightforwardly applicable to other image-based telemedical applications, for instance teledermatology.

References

[1] S.M. Hudson, R. Contreras, M.H. Kanter, S.J. Munz, D.S. Fong, Centralized Reading Center Improves Quality in a Real-World Setting. Ophthalmic Surg Lasers Imaging Retina. 2015;**46**(6):624-629. doi:10.3928/23258160-20150610-05. PMID: 26114842

[2] G. Lotz, T. Peters, E. Zrenner, R.Wilke, A domain model of a clinical reading center - Design and implementation. Conf Proc IEEE EngMed Biol Soc. 2010. 2010:4530-3. doi: 10.1109/IEMBS.2010.5626032.

[3] I.V. Ivanov, M.A. Leitritz, L.A. Norrenberg LA, M. V¨olker, M. Dynowski, M. Ueffing M, J. Dietter, Human Vision-Motivated Algorithm Allows Consistent Retinal Vessel Classification Based on Local Color Contrast for Advancing General Diagnostic Exams. Invest Ophthalmol Vis Sc. 2016; **57**:731-738. doi:10.1167/iovs.15-17831

[4] K. Atalag, T. Beale, R. Chen, T. Gornik, S. Heard, I. McNicoll, openEHR: A semantically-enabled vendor-independent Health Computing Platform, White paper, http://www.openehr.org/resources/white_paper_docs/openEHR_vendor_independent_platform.pdf, last accessed on Nov 07 2016

[5] R. Daigneau, Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services, Addison Wesley, Boston, 2012

[6] S. W. Ambler, The Design of a Robust Persistence Layer for Relational Databases, White paper, http://www.ambysoft.com/downloads/persistenceLayer.pdf, last accessed on Nov 07 2016

[7] M. Fowler, Patterns of Enterprise Application Architecture, Addison Wesley, Boston, 2003

[8] http://basex.org