Clinician-Led Development of Electronic Health Records Systems

John CHELSOM[a], Raju AHLUWALIA[b], and Naveed DOGAR[a]

[a] *Centre for Health Informatics, City University, London*

[b] *NHS, NW Thames Orthopaedic Rotation*

Abstract. The open source cityEHR system was evaluated as a toolkit for clinician-led development of an electronic health record for management of patients in the Ponseti clinic of a major London hospital. As a toolkit, it was found that the ontology-driven approach of cityEHR was too complex for clinicians to use. The toolkit was refined to use more familiar spreadsheets to represent the ontologies and was then used successfully to create an effective clinical system, generated automatically from the information model.

Keywords. Electronic Health Records, Open Standards, Open Source, Clinician-led Development

Introduction

The cityEHR electronic health records system has been developed as an integration of 'best of breed' open source software components, configured to support open standards for clinical records such as ISO-13606 [1] and HL7 CDA [2]. It provides clinical users with a toolkit that enables them to develop their own Electronic Health Records (EHR) systems and then to deploy them on an enterprise-scalable platform.

The system was originally conceived to address the fundamental issues that caused the failure of the EHR systems implemented as part of the National Programme for IT in England [3].

The platform for the deployed EHR is shown in Figure 1. It combines major open source software components that run on an Enterprise Java platform and that are freely available on the web: Apache Tomcat (tomcat.apache.org), eXist (exist-db.org), Mirth Connect (www.mirthcorp.com), Orbeon (www.orbeon.com) and Fusion Charts (www.fusioncharts.com). These components are integrated and configured using the XML-based languages XForms, XSLT and XQuery; there is no compiled code in the cityEHR.

The EHR deployed on this platform is configured entirely from an information model represented as an ontology, in the Web Ontology Language (OWL) [4]. This ontology uses a core architectural model (represented by OWL classes, object properties and data properties) as the building blocks for the information model that contains specific individuals and property assertions.

This OWL information model is then transformed into a set of XML configuration files that define how the (automatically) generated cityEHR system functions for clinical users. The main XML representation in this configuration is the XForms language for specification of clinical data entry forms, summary views and letter templates. Behind the XML configuration of the user interface lies the underlying model of the patient data which uses the XML-based Clinical Document Architecture (CDA) from HL7.

Thus there are four stages to the generation of patient data in the functional cityEHR system: architecture, clinical information model, runtime configuration, stored data in the clinical system [5].

The purpose of this study was to evaluate the cityEHR toolkit by developing a system for recording patient encounters at the Ponseti out-patients clinic at a major hospital in London, UK. The main research question to be answered was: 'Can clinicians develop their own information model that can be used to automatically generate a functional EHR system?'

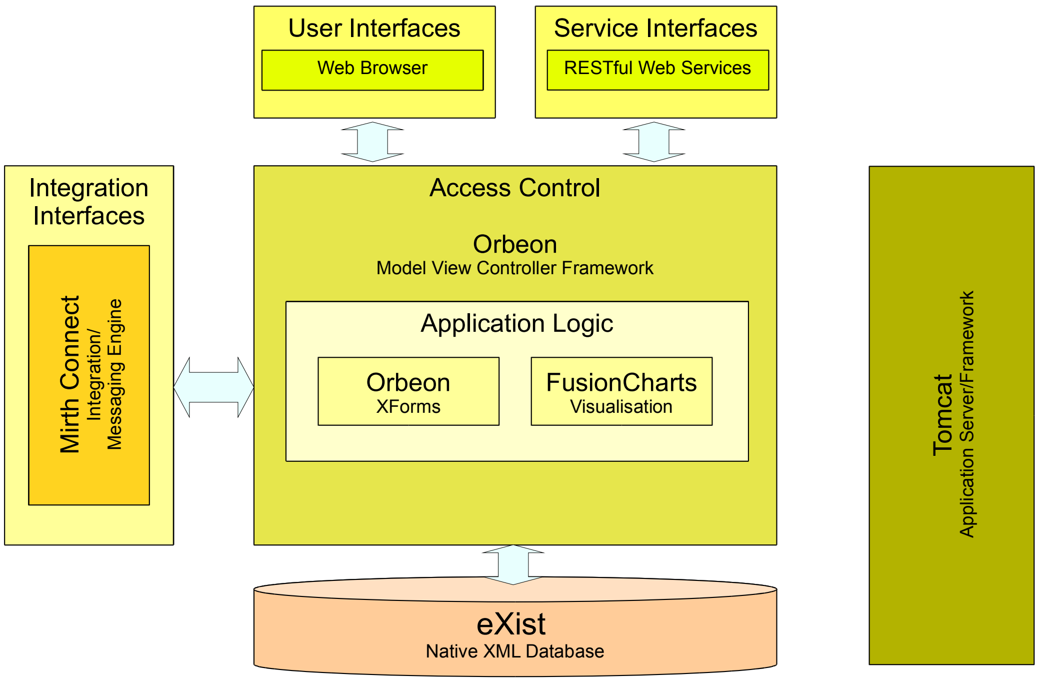

Figure 1. cityEHR Reference Platform of Open Source Software Components

Requirements

The requirements for the Ponseti system were gathered without reference to the capabilities of the cityEHR system or modeling toolkit. A clinical analyst worked with clinical staff (an orthopaedic surgeon and specialist clinic nurses) to gather requirements using structured interviews and analysis of existing paper-based data records.

The products of this analysis were a set of Use Cases, a System Requirements Specification, a Data Dictionary and a set of User Interface Wireframes. The base functional requirements were relatively simple and in our experience are typical of the requirements for most systems for outpatient clinics and research studies.

Clinical Data Collection	**Other Requirements**
Edit patient data systematically and store in a database	Secure authentication method to verify users
Clean, functional, form based user interface	Basic role-based access control
Pre-defined data sets in a drop-down menu	
Forms clearly sectioned into given headings	Complex queries on the patient database
Attach/upload multimedia to the patient record	Output resultant queries in a report
Conditional inclusion of sections/entries on forms	Graphical or/and tabular format for reports
Enforce mandatory fields for data entry	Display summary view(s) for existing records
Hints - short help assistance with data entry	Generate standard letters (e.g. in MS word)
Hyperlinks to external web resources	Query to assemble cohorts for research studies
Calculations and scoring on assessment forms	
Automatically calculate standard data (e.g. patient age)	

Implementation Using cityEHR

The basic functional requirements and user interface wireframes were used as the starting point for configuration of the cityEHR.

The data model elicited from clinical users was coded as an ontology by the clinical analyst, using the open source Protege tooling from Stanford University [6]. Figure 2 shows the user interface of Protege with the basic architecture of the cityEHR represented as classes and the specific information model for the Ponseti clinic system represented as individuals, object property assertions and data property assertions.

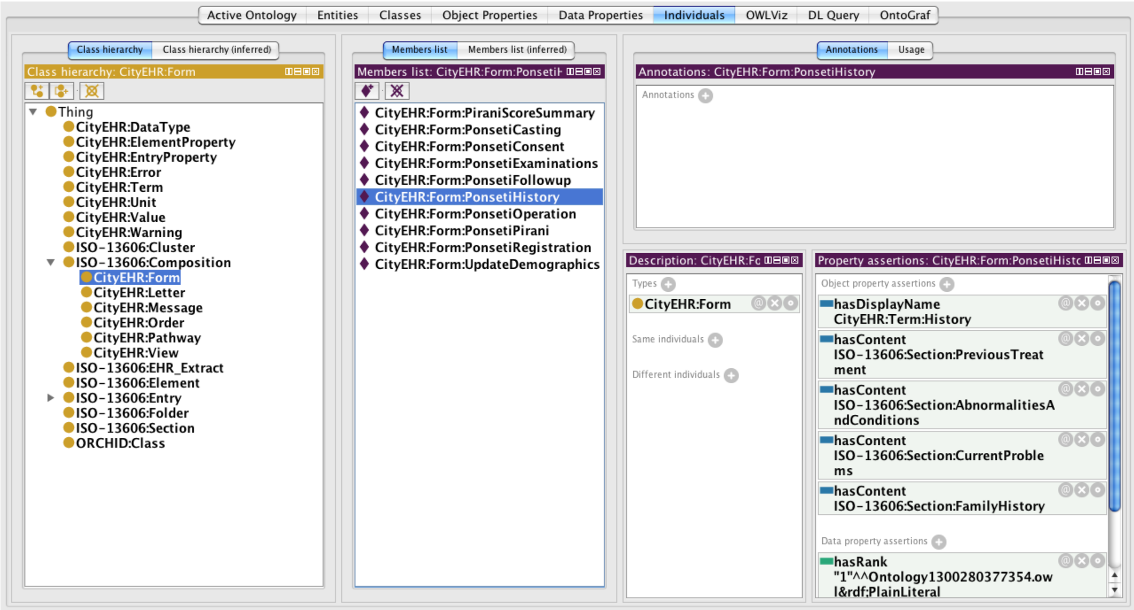

Figure 2. Information Model as an Ontology, Edited Using Protégé

The resulting OWL model was then loaded to the cityEHR system and generated a working EHR prototype which was compared with the original wireframes and functional specification.

It was found that the user functionality and general layout of data entry forms was created to the general satisfaction of users, but that a number of specific functional requirements were not met by the automatically generated system, specifically:

- complex calculations and scoring on assessment forms
- calculation and representation of patient age in days, weeks, months or years depending on the age of the patient
- attach/upload multimedia to the patient record

These features were then added to the cityEHR system in a two stage process: firstly adding the semantics to the cityEHR architectural ontology so that the features could be specified in the information model and secondly adding functional support for the features when the EHR system was generated from the model. Having added these features to cityEHR it was then possible to generate a clinical EHR system that was acceptable for piloting in the Ponseti clinic, as shown in Figure 3.

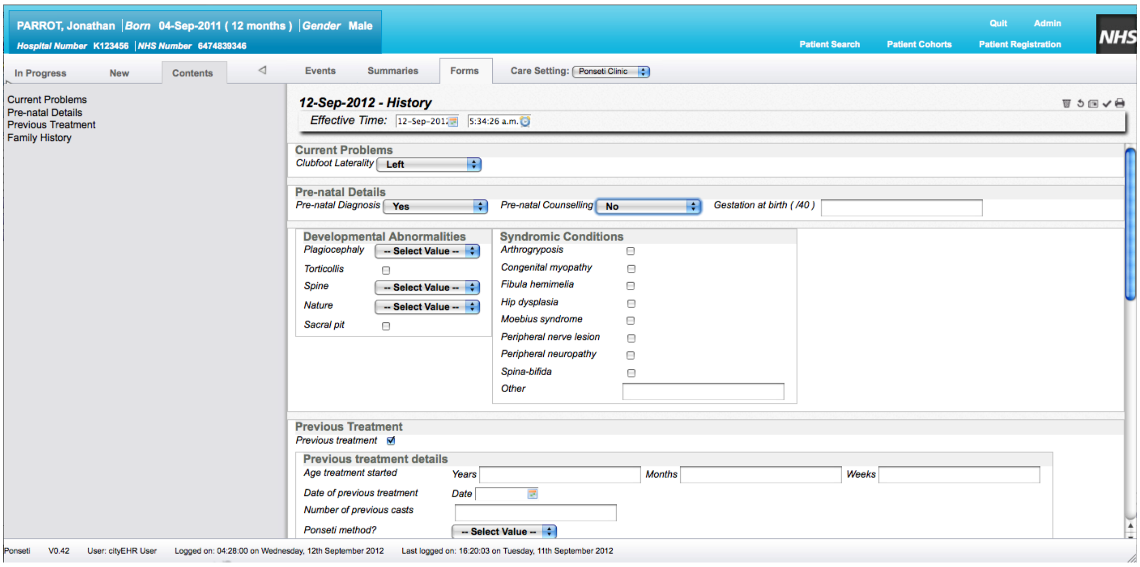

Figure 3. Electronic Health Records System Generated from Information Model

Spreadsheets for Clinical Users

Although it was possible for the clinical analyst to create an information model using Protege and use that model to generate a clinic system in the cityEHR that met the functional requirements, there were some major issues with the usability of the toolkit.

As the information model increased in size the clinical analyst struggled to create a consistent model in a timely fashion; it proved impossible for clinical staff to use the Protege tooling themselves. The three main issues encountered were:

- The atomic assertions of the OWL language mean that even simple constructs (e.g. the title of a section in a form) require two or three levels of abstraction in the model; as the model grows, it becomes very difficult for a user to work with this level of abstraction, even using a graphical tool such as Protege
- Although reasoners can be run on the OWL model to check its consistency, the language (and the Protege tooling) does not provide easy mechanisms for constraining user input so that consistent models are made in the first instance.
- For clinical users, unfamiliar with the concepts of ontology and with the Protege tooling, the learning curve proved too steep to allow them to work effectively within an already busy clinical schedule.

To address these three issues, the toolkit was 'refaced' to use a standard spreadsheet to specify the model. The spreadsheet follows the structure of ISO-13606 and allows constraints to be placed on the data entered into various fields. The spreadsheet can then be saved as XML and transformed using XSLT into the OWL representation of the model.

Figure 4 shows a sample of the spreadsheet tooling that was developed using Open Office Calc (the open source equivalent of Microsoft Excel).

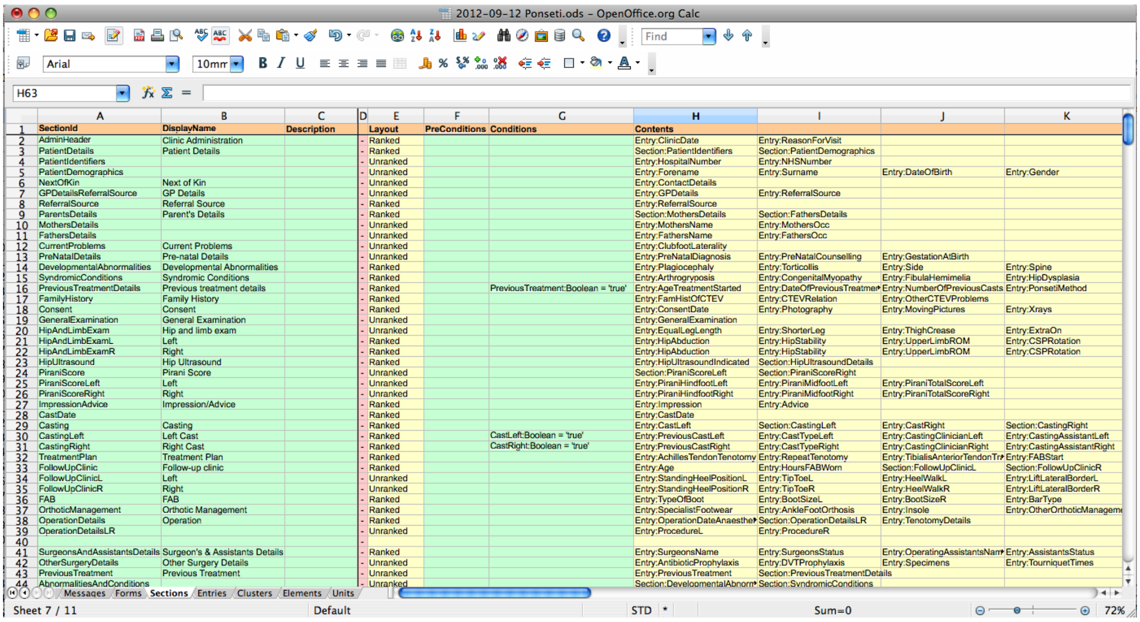

Figure 4. OpenOffice Calc Spreadsheet Mirrors the OWL/XML cityEHR Architecture

Conclusions

Our study in the Ponseti clinic proved that is is possible to generate a usable EHR system directly from an information model specified by clinical users.

However, it also showed that the complexity of the ontology-based approach (both conceptually and in the available open source tooling) made is impossible for clinicians to develop that model themselves directly using the cityEHR tools.

Once a simpler spreadsheet-based interface was introduced to specify the information model then it did become possible for clinical user to create and modify their information models directly.

In subsequent studies we have found that a variety of tooling can be used to create the models (not just spreadsheets). This more general approach is shown in Figure 5.

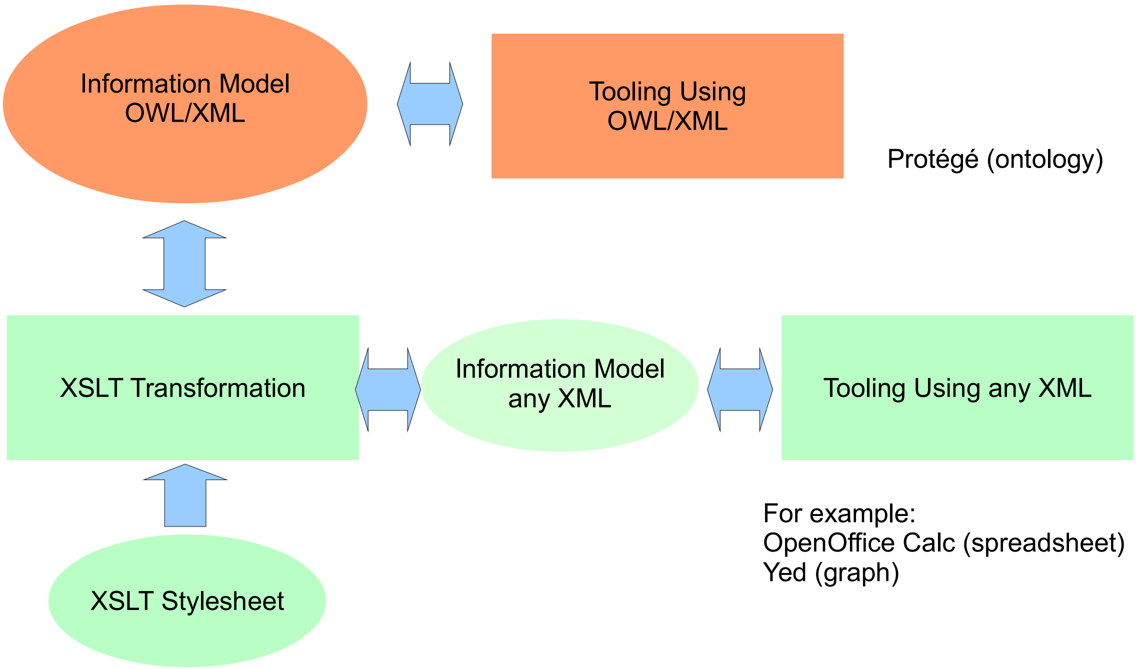

Figure 5. Information Model in OWL/XML Transformed to any XML for Editing

The openness of the OWL/XML language, the widespread support of XML across modeling tools and the capabilities of XSLT for transformation mean that this approach provides a flexible approach to information modeling that can be used by clinical users to automatically generate functional EHR systems using open source platforms such as cityEHR.

References

[1] ISO 13606-1:2008 Health informatics - Electronic health record communication - Part 1: Reference model. Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40784

[2] Dolin RH, Alschuler L, Boyer S et al. HL7 Clinical Document Architecture, Release 2. J Am Med Inform Assoc. 2006;13(1):30-9.

[3] Chelsom, J. Open Health Informatics–A Fresh Approach for NHS IT. City University Challenge Paper (2011). Available at: http://www.openhealthinformatics.org

[4] World Wide Web Consortium. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. Available at: http://www.w3.org/TR/owl-features/

[5] J. J. Chelsom, I. Pande, R. Summers, I. Gaywood. Ontology-driven development of a clinical research information system. 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 1-4, 2011 24th International Symposium on Computer-Based Medical Systems, 2011

[6] John H. Gennari , Mark A. Musen , Ray W. Fergerson , William E. Grosso , Monica Crubzy , Henrik Eriksson , Natalya F. Noy , Samson W. Tu.The Evolution of Protégé: An Environment for Knowledge- Based Systems Development (2002) International Journal of Human-Computer Studies.