MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Intelligent Querying of Terminology Services: A Web-Based SNOMED© CT Servlet

Ignaz FA Reicht[a], Sebastian Garde[b]

[a] *University for Health Sciences, Medical Informatics and Technology, UMIT Hall in Tyrol, Austria*
[b] *Central Queensland University, Health Informatics Research Group, Melbourne, Vic, Australia*

Abstract

Clinical terminologies are playing a major role in realising Electronic Health Records (EHR). These terminologies are becoming increasingly complex and comprehensive to enable more detailed recordings of the patient. This may increase quality of documentation but is also time-consuming and error-prone if a clinician has to navigate through a "jungle of codes" as this is often the case in current systems. The purpose of this paper is to show how the power and features of terminologies can be made accessible to the clinician. For this we developed a dynamic JavaServlet application, which provides access to intelligent SNOMED-CT search queries registered with Ocean Informatics' Terminology Server. The architecture of the Servlet and provided Webservices will enable interactivity between user and Terminology and thus reduce the time and effort required by clinicians when coding using clinical terminologies. As our JavaServlet application demonstrates, user-friendly querying of complex terminologies – even if they constantly change and are getting more complex – is possible. Our service-oriented, web-based architecture appears to be reasonably stable and sufficiently fast.

Keywords:

SNOMED-CT, terminology, Electronic Health Records, computerized medical record systems, openEHR, archetypes

Introduction

Electronic Health Record (EHR)

The characteristics of EHRs are in general as follows:

- Patient-centered
- Longitudinal, this means a long term record of care would be preferable, in the best case from the cradle to the grave
- Comprehensive, which includes records of all care events and all relevant measurements
- Prospective, this provides instructions and goals in healthcare for the patient.

To implement these EHR characteristics Information Technology (IT) is a powerful instrument. It enables efficient communication supporting a high level of Data integrity – once recorded available anywhere – Information must be of high quality, correct, complete, reliable and flexible. However, without interoperability, correct and trustful interpretation of the stored data might lead to more complex and variable processes imposing additional workload and sources of errors on clinicians [1].

A systematic collection of patient relevant data including what, where, how, when, why and who took and takes care of the patient will empower the quality of patient care. But it depends on how clinicians and carers can handle the provided IT systems, and thus on their usability. If the symbiosis between them increases, both, clinicians and IT systems will benefit from higher quality. Clinicians will make serious usage of IT features, and thus symbiosis and evolution will continue.

During the last decades there have been huge efforts in research and development of terminologies. To fulfill desired requirements of users, clinical terminologies have become more complex and comprehensive. There is a need to create applications, which can enable the usage of complex and comprehensive features. To our knowledge usability of complex and comprehensive clinical terminologies is quite difficult due to current IT solutions do not consider socio-technical aspects very well.

The aim of this paper is to

- Briefly identify common problems with clinical terminology
- Describe the development of a webbased application (JavaServlet), which enables access to intelligent queries implemented on a SNOMED-CT Terminology Server and thus overcomes some of the identified problems with clinical terminologies.
- Discuss how this use of clinical terminologies fits together with the use *open*EHR archetypes as models of clinical content.

Materials and methods

To reach our goal, we had to investigate available resources, instruments and which applications and tools are compatible for our purpose. From a technical point of view, we tried to find the most innovative solution with the best performance for the development of the application in discussions together with members of Ocean Informatics, by studying specifications and manuals and developing prototype systems to experiment with specific approaches.

*open*EHR *and Archetypes*

For future-proof development of EHRs representation, context and semantic integrity are essential. The *open*EHR and Archetypes approach can cope well with this issue. *open*EHR is originally based on the GEHR projects (Good European/Electronic Health Record). The aim is to support the communication and availability of routine data and development of EHR. It focuses on data and semantic interoperability of complete EHRs. An Archetype is a smart specification which can represent one complete clinical concept by separating clinical data collection from the record keeping concerns. E.g. a blood pressure archetype represents a description of all the information a clinician might want or has to report about a blood pressure measurement [5].

Ocean Terminology Service (OTS)

According to the product specification OTS, consists of three components, Terminology Service, Terminology set query editor and application widget for clinical application use. The terminology service restricts the set of possible terms used to populate a given data field based on the constraints expressed in archetypes - while allowing browsing of the terminology within this knowledge environment. OTS is highly scalable and supports the comfortable usage of complex terminology in clinical areas [6] in our case SNOMED-CT. In this paper we use OTS to access SNOMED CT queries via webservices.

SNOMED-CT Model

The SNOMED-CT concept model supports both pre- and post-coordinated concept expression. Post-coordinated means, that an expression which is sufficient to define a concept representation but which has not been assigned a name by being used in a definition. Pre-coordinated is a named concept representation, which is either defined, or not. The fundamental design of the SNOMED CT concept model refers to the way in which the concepts in SNOMED CT are defined and related to one another. This includes features and limitations about concepts, relationships, characteristics and roles. Within the constraints of the Fundamental Concept Model Design, the way that particular attributes are applied has a significant impact on the way that SNOMED-CT concepts are defined and the ability to compare them. This includes the use of specific sets of released attributes for findings, context model attributes providing additional defining characteristics for context dependent concepts and role hierarchies [7].

Results

In the following, we will describe the mechanisms our Servlet uses and the concepts it is based on. These mechanisms solve typical problems inherent in many clinical terminologies as the following brief literature review about typical problems in clinical terminologies reveals.

Terminology

Clinical terminology concerns the meaning, expression, and use of concepts in statements in the medical record or other clinical information systems. During the last decades there have been huge efforts solving problems of standardizing medical language and terminology but it is unknown if the development and research will ever succeed in this case [2]. In this paper, Rector also mentioned 3 assumptions about the context in which terminologies are being used:

- The purpose of clinical terminology is to support clinical software
- All terminologies will have to support conversion to existing reporting and epidemiological coding schemes
- All terminologies will need to be multilingual

The more specific the terminology, the more complex and comprehensive its vocabulary in a given area. E.g. there are two kinds of burn – thermal, chemical – and 200 body locations, using pre-coordination we would get 400 codes for burns. If we add 3 thicknesses plus unspecified we have 1.600; if we add three extents plus unspecified we have 6.400; add three levels of recency plus whether or not the burn is complicated by infection the total amount will rise to 76.800 codes only for burn. Another problem is a fundamental conflict between the needs of users and the requirements for developed software. The way humans and machines process information is very different. Machines with their digital processing capabilities are intolerant, fixed and not flexible compared to human beings with analog data processing [2].

Above we have focused on the issue of "what" the terminology can express, but what is about "how" the terminology can express its content and how the vocabulary is organized in it. Cimino [3] describes 2 approaches. One is a systematic case description by choosing the suitable term. A Fracture for example – in the first round you choose between simple or compound, in round two you decide between greenstick, avulsion, compression etc. This avoids nonsensical combinations and avoids the large numbers of terms occurring through the explosion of combinations.

Another approach will allow the user compositional extensibility by combining all the atoms of the terminology into necessary coded terms. The problem within this alternative is, that the atom cannot only consists of its letters (e.g. the word / atom *"White"*) there is a need to add a characterization. For example with the atoms *"White"* and "*Conjunctiva"* you can formulate the term "*White Conjunctiva",* but how would you formulate "Wolff-Parkinson-*White* Syndrome"? [3]

By reading medical informatics research, Terminology-systems tend to dealing with the notion of concepts. That means, that each individual term refers to an individual meaning [3].

Afterwards many vocabularies and terms are organized into strict hierarchies. For example a concept with the unique identifier/code 1000 might be the parent of the concept with the code 1200 which might be the parent of the concept 1280 and so on. With computer interfaces querying for members of a class or concept become easier (search for "all codes beginning with 1" will retrieve codes 1000, 1200, 1280). But within a polyhierarchical terminology where one concept can also belong in more than one location in the hierarchy, a single hierarchical identifier is no longer possible. Therefore is it desirable, to have the unique identifiers for the concepts which are free of hierarchical or other implicit meaning. [3]

Now search routines have to be smart enough to cope with these complex hierarchical and concept identifiers. Nevertheless, front-end terminology applications (like terminology browsers) have to be simple, easy to handle and smart enough to identify what information the user requires in what context.

Terminology browsers allow users to search and navigate through the terminology without necessarily providing editing capabilities. E.g. MicroMeSH was one of the first tools created for interacting with complex standard terminology. This program provided a search assistant with a rich set of lexical look-up facilities and a tree walker to support navigation through terminology.[4]

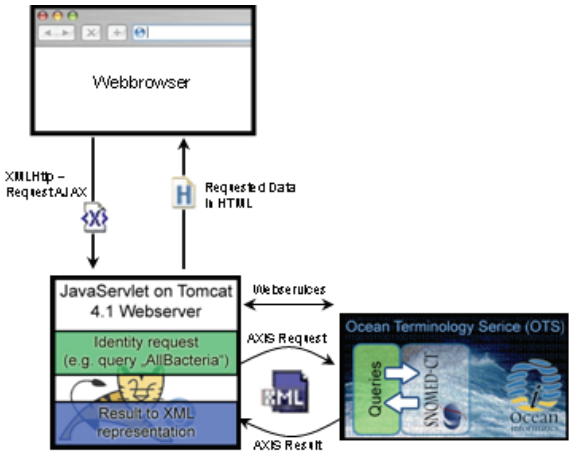

Figure 1 - Overview of the system architecture

After we made us familiar with typical problems of clinical terminologies for our development solutions and concepts have to be found, which overcome some of the problems mentioned above and can lead to innovative results. For development and research in this case, problems in interoperability have to be solved and the openEHR Archetype approach can lead to solutions. As the openEHR and archetype approach does not rely on big standardized terminologies but on "micro-vocabularies", it offers more flexibility during standardization and translation of clinical concepts and overcomes some of the shortcomings of terminology-focused approaches. Terms within an archetype can be bound to or constrained by any number of terminologies or, if highly specific to the archetyped concept, be specified within the archetype itself. To SNOMED-CT, archetypes provide the required structure for clinical concepts. Each term can be bound to a SNOMED-CT concept to specify explicitly that the archetype term and the SNOMED-CT concept have the same meaning. In addition each archetype term that is a text can be constrained by a terminology like SNOMED-CT, ICD etc., e.g. the allowed values (value set) for a diagnosis text in the archetype could be constrained by the appropriate subset of a terminology that deals with diagnoses. Archetypes in contrast to terminologies specify groups of data that as a whole are whole, discreet, highly related and clinically meaningful concepts like a blood pressure measurement consisting of the diastolic and systolic blood pressure, but also of any other information relevant for the interpretation of the readings, e.g. the position of the patient during the measurement [1].

The Servlet

In general the Servlet represents an interface between user and implemented Webservices for querying SNOMED-CT on OTS (Figure 1). The Servlet can manage two kinds of incoming data. First, information about desired commands

or queries, which will be handled within the application so desired processes can take action on OTS. In our case Webservices enable queering SNOMED-CT will be processed. After calling, OTS delivers results to our Servlet application which interpret and represents these in HTML via Webbrowser. As a consequence of this design, Bacteria for example, which are defined in Snomed CT at various locations, can be pulled together and presented coherently for a specific purpose.

It was quite difficult to access our desired information because of the abundance of different technologies and standards. In the following we present an approach to realize such an interface. The Webservices on OTS are based on .NET® and they are SOAP1 and SOAP2 compatible (SOAP: Simple Object Access Protocol) With SOAP you can exchange information between different systems. This technology is comfortable because it provides platform independency. The existing Archetype Finder (http://www.archetypes.com.au) already provides Apache AXIS1 based Webservices (Axis: Java and XML based Webservice framework, comfortable for creating interoperable applications). A couple of other specifications and tools like AXIS2, XINS (XML Interface for Network Services), JWSDP (Java Web Services Development Pack) and XFire were investigated, but AXIS1 was the easiest way to fulfill our motivation with a stable suite of webservices and we therefore used AXIS1 Technology also in our application. In addition, we chose a Tomcat 4.1 Webserver with JRE 1.4 (Java Runtime Environment).

AJAX(Asynchronous JavaScript + XML) and servlet functionality

To support high performance during querying the AJAX Technology provides the most suitable transmission architecture in our case. The big advantage of this Technology is in the *"XMLHttpRequest"* Object, which is responsible for transport of commands or actions a user wants to run on the Servlet. The Servlet connects to the Webservices on OTS and lists all available queries. The requested Data collected by the processed query will be represented as HTML by the Webbrowser (Figure 2).

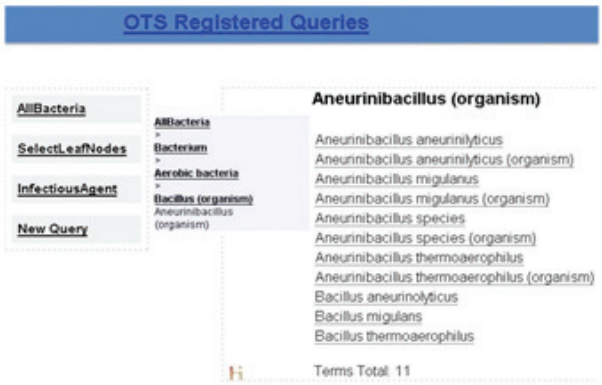

Figure 2 - The servlet in a web browser

To call a command on OTS for example to list all bacteria, we needed the definition of the Webservices on OTS. For programming the Servlet to transmit the required commands and information, the provided WSDL-file on OTS has to be converted to Java Classes by the open source tool CASTOR (so that the Servlet can take action). Now the "highway" to the Webservices is prepared, but we do not know what kind of vehicles (datasets) can drive on it to reach the Webservices. OTS defined the datasets by XML-Schemas (XML: Extensible Markup Language). The Webservices accept two different datasets: registered queries and defined terms. To implement and prepare the Servlet, CASTOR needs to be used again. After all these conversions the Servlet is able for correct interactivity and interpretation with the Webservices registered on OTS. Figure 3 shows all available queries. For example "*AllBacteria*" is a registered Query, which lists all Bacteria within the SNOMED-CT. All entries are grouped, so comfortable top-down navigation between the levels is given. First we can choose between different types of Bacteria. For example if we want to know all available entries of "Aerobic bacteria" (Figure 4) the AJAX engine will deliver a command to the Servlet to go one level deeper into "Aerobic bacteria". If you want to navigate one ore more levels higher AJAX will transmit a "go up" command to the Servlet. As you can see with AJAX is it possible to query only the requested data, this means the Servlet has not to re-query and rebuild already presented data and information.

These search routines, queries and navigation of the Servlet in cooperation with OTS Webservices will deliver fast and situation–specific results for clinicians and care professionals.

Discussion

The more detailed and specific terminologies are, the more complex and less comprehensible they get. Research of

terminologies provides huge potential for new and innovative approaches, but actually we are not able to define what will be the most suitable direction for the future [2]. However, we have to find ways and solutions to enable all features provided by terminologies, therefore considering socio-technical aspects of usability will lead to future orientated results. Without intelligent search strategies and navigation concepts, we believe that growing clinical terminologies are no longer handable leading to suboptimal quality of documentation. To prevent a lack of quality *open*EHR decided in cooperation with Ocean Informatics to construct services and applications, which can cope with the evolution of terminologies. As mentioned in results the *open*EHR archetype approach offers more flexibility and overcomes some of the shortcomings of terminology-focused approaches ([1], [5]).

Further, *open*EHR and Archetypes grant interoperability achieved by the 3 level definition. The *open*EHR reference model alone ensures syntactic (data) interoperability. Structural interoperability is achieved by the definition and use of archetypes. E.g. a blood pressure archetype will have the same meaning no matter where or in which EHR it appears. The third level describes semantic interoperability. In addition to archetypes and the reference model, archetype development must be coordinated through systematic governance. Incompatible, overlapping archetypes expressing the same concept for example have to be avoided[1].

An archetype specifying SNOMED-CT terms by including our Servlet is now available in EHR and can share its clinical knowledge and information can be safely interpreted by exchanging archetypes.

In our case all employed tools and programs to generate our Servlet served the purpose and provided correct and reliable results. The Servlet is running stable. The implementation of the AJAX Engine supports a convenient and easy to use search routine. It works quite similar to the *"GoogleSearch bar"*. If your cursor is in the search field and you are typing an "A", a list of available criteria starting with "A" will be shown nearly instantly.

Figure 3 - Registered Queries

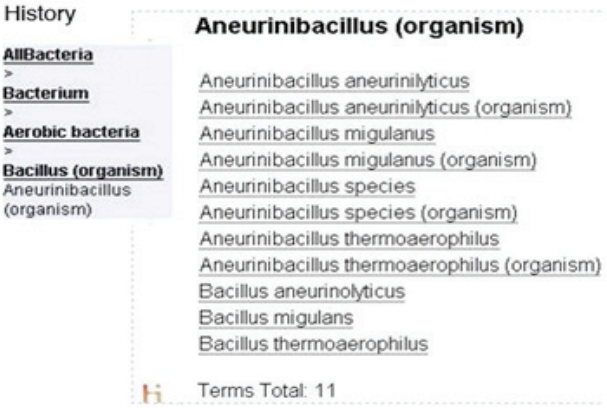

Figure 4 - JavaServlet querying "AllBacteria" on OTS; History box lists visited subcategories of "AllBacteria"; underlined terms processes to go down one more level and list terms

This proof of concept shows that the Servlet can enable available Webservices and queries on OTS and by these means improve the usability of clinical terminologies. The advantages of this approach are the following: First, the Servlet is flexible and dynamic, so there is no need to alter any parameter if queries are changing or moving. The second advantage is due to the web-based concept with a service-oriented architecture, which enables effortless integration into EHR applications. Together, this eliminates the need for directly integrating (hard-coding) clinical terminologies into software and the need for upgrading the software when the clinical terminology changes.

With regard to performance, to make our approach future proof, we need to explore the following approaches:

- Caching Results
- Cache Clustering (locate already loaded Data in cache of other users)

- Server Mirroring

The next goal will be the transport of required data into EHR applications. There is a huge range of possibilities and we will do research to find the most suitable one.

Conclusion

By means of our Servlet we were able to demonstrate that user-friendly querying of complex terminologies – even if they constantly change and are getting more complex – is possible. The technologies chosen for this purpose (Java Servlet technology, AJAX, AXIS in a service-oriented, web-based architecture) appear to be reasonably stable and sufficiently fast.

References

[1] Garde S, Knaup P, Hovenga E, Heard S. Towards Semantic interoperability for electronic health records: domain knowledge governance for openEHR archetypes. Accepted for *Methods Inf Med* 2007.

[2] Rector AL. Clinical terminology: why is it so hard?. *Methods Inf Med* (1999) 38: 239-252.

[3] Cimino JJ. In defense of the desiderata. *J Biomed Inform* (2006) 39: 299-306.

[4] Cimino JJ. Terminology tools: state of the art and practical lessons. *Methods Inf Med* (2001) 40: 298-306.

[5] Garde S, Knaup P, Schuler T, HovengaE. Can openehr archetypes empower multi-centre clinical research?. *Stud Health Technol Inform* (2005) 116: 971-976.

[6] http://www.oceaninformatics.biz. (last visit 4 Dec 06)

[7] http://www.snomed.org/clinical/documents/CMWG_ScopeandTermsOfReference20041231-ks-comments.pdf (last visit 4 Dec 06)

Address for correspondence

Ignaz Franz Andreas Reicht
Griesbachweg 46
A-6370 Reith bei Kitzbuehel
Ignaz.reicht@umit.at