Ruby implementation of the openEHR specifications

Shinji Kobayashi[a], Akimichi Tatsukawa[b,c]

[a] *Department of Bioregulatory Medicine, Ehime University Graduate School of Medicine, Ehime, Japan*
[b] *Narimasu Kosei Hospital, Tokyo, Japan*
[c] *Department of Medical Informatics and Economics, Graduate School of Medicine, University of Tokyo, Japan*

Abstract

The openEHR project has developed the specification for future-proof interoperable electronic health record (EHR) systems. This project provides the specification and implementation on which the ISO/CEN 13606 standard is based. The implementation has been formally described in Eiffel, C# and Java, but not in scripting languages. A team from Japan has implemented this specification using the Ruby language for the efficient development of new healthcare computing environments and for investigation of the universal applicability of the openEHR specification.

Keywords:

Electronic health records, openEHR, ISO/CEN 13606, Ruby, Open-source software.

Methods

The Ruby implementation of openEHR was developed with Ruby 1.8.6 or later.

More information on the Ruby implementation project and the source code are available at http://openehr.jp/ref-impl-ruby/. In addition, these resources are available under the openEHR open source software license (Mozilla tri-license) like the other implementations of the openEHR project.

Because our implementation policy involves agile programming, we established the rule that unit tests should be written before working code.

Results

We implemented most of the openEHR specification and Archetype Definition Language (ADL) parser. We also implemented RESTful terminology server experimentally.

ADL Parser

An adl_parser is an implementation of ADL parser library in Ruby programming language. It is built on Racc, a LALR(1)-type parser generator that is bundled with the Ruby standard library. Currently, we are investigating a design and implementation of these facilities to fit with the specifications correctly.

Reference models (RM)

RM is used to describe health record itself in the openEHR. We implemented the most of all specification. Some specifications are not yet implemented in other language implementation. Because some packages were not determined specification in detail, we could not implement such packages that even the other projects had not implemented yet. For example, RM::Support::Measurement, RM::DataTypes::TimeSpecification were not implemented in Java or Eiffel.

Archetype Object models (AOM)

AOM is a runtime image of archetype in openEHR system. We also implemented these class packages.

RESTful terminology server

We designed RESTful API as web service. An example query is shown as bellow URI (1).

http://localhost:3000/terminology/openher/en/1 (1)

This URI means protocol, server address, terminology access, category, encodings, code for terminology. This request returns simple XML format terminology information.

Conclusion

We implemented most of the openEHR specification using Ruby and have encountered some complications in doing so. Because the Ruby language has the capacity to solve such problems, the Ruby implementation of openEHR may provide a modern and practical platform for healthcare information systems.