e-Zest

TRANSFERRING PATIENT RECORDS BETWEEN ELECTRONIC HEALTH RECORD SYSTEM'S USING ATL

Application Overview

The application serves as an electronic health record system for organizations that deliver healthcare as a service, such as hospitals and physicians. The system provides services to healthcare practitioners by providing them one stop solution for managing the patient information in a user friendly way.

The application caters to the security concerns by displaying records through different levels such as therapeutic relationships, confidentiality and getting the latest information through government managed servers.

The Vision

The aim is to develop a platform which can be used to develop specific applications which can be used for individual hospitals, smaller clinics or physician offices.

Technologies and Tools

The application is a cloud-based EHR software. It is written in Java, with a JAX-RS server-side REST API, and a JavaFX rich client that supports online/offline operation and secure data synchronisation. It also has a mobile client, written as a single-page HTML5 / JavaScript app that communicates directly with the REST API via CORS.

Other Tools used for application development are Eclipse, Jira, Wiki, SharePoint, Jenkins, Git, and Bitbucket

Problem Overview

EHR systems need to provide support to transfer the Patient medical history to other EHR systems. The transfer of Patient records is generally required in scenarios where in the Patient needs to change his/her regular Caregiver or consult another Caregiver. In such scenarios it is desirable that the Medical history of the Patient be made available with the new Caregiver.

To facilitate the transfer of data between EHR systems, the Health authorities in various countries generally define a standardized XML format .The XML formats are further made available in various sub formats to provide transfer of:

- Basic Patient Medical history.
- A more detailed Patient Medical history including instances of when the Patient visited the Caregiver.
- Complete Patient History including all changes done to the Patient records.

EHRs need to provide support to Import and Export Patient Medical data in all the versions of the standardized formats.

Challenges and Solutions

- The client EHR domain model is based on Hibernate whereas the format defined by Health Authorities is XML based.
- The XML schema is too generic whereas the EHR domain model is designed based on the application requirement.
- Mapping between EHR domain model and XML model involves complex rules and conversions.
- The mapping logic is intermingled with calls to DB to lookup Master database.
- The XML files are in several MBs and the conversion logic needs to work fast.
- There are lots of Entities and rules involved. Determining each mapping rule in advance is very tedious. The mapping logic had to be reviewed by a functional expert to confirm correctness during the development stage itself.

Solution Architecture

We started writing the conversion logic using plain Java, JAXB and Dozer mapping but we soon realized that this approach resulted in an explosion of custom Dozer mapping classes, the mapping code was mixed with the DB calls and conversion logic. It was also very difficult to get the mapping logic reviewed from a functional expert. As we progressed further the need for a better transformation solution was increasingly felt.

The EHR solution was already based on Model Driven Engineering and was using EMF based models for code generation. Upon reviewing the previous approach and the benefits provided by the ATL technology, a decision was made to use ATL for transforming the data between the XML and EHR models.

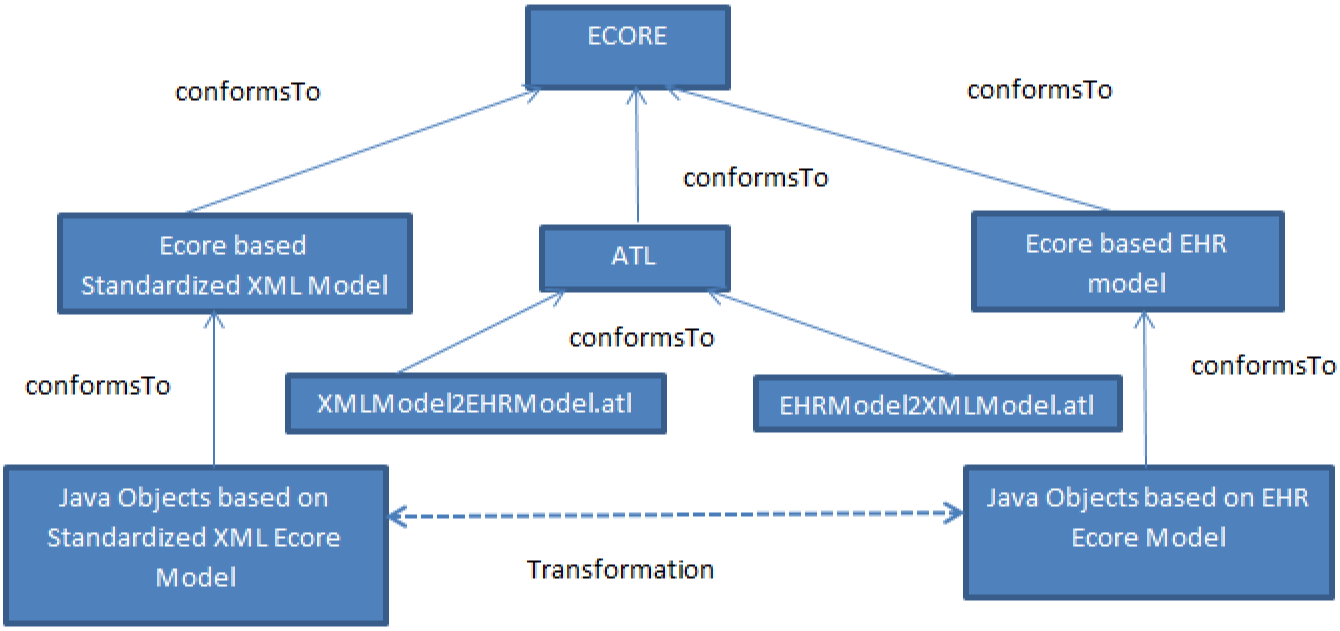
ATL

ATL (ATLAS Transformation Language) is a model transformation language and a toolkit. In the field of Model-Driven Engineering (MDE), ATL provides ways to produce a set of target models from a set of source models. An ATL transformation program is composed of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models.

Developed on top of the Eclipse platform, the ATL Integrated Environment (IDE) provides a number of standard development tools (syntax highlighting, debugger, etc.). These aim to ease development of ATL transformations.

XML and EHR model Transformation Steps

- As a first step the XML EMF model and the EHR EMF model are derived from the XML schema and the EHR hibernate based domain classes respectively
- The Lookup data and the Input model are provided to the ATL program through a Java program
- The transformation logic is written in ATL files which analyses the data in the Input model and performs the transformation and conversions. It also uses the Lookup data to perform the transformations
- The output model is then used in the Java program to either persist the data in database for impo rt or to export the output model to XML files in case of Export



Benefits of the solution

- The mapping logic is clearly separated from the DB calls
- All the mapping logic is available at a single place
- Complex rules can be written easily using ATL
- The ATL language is easy to read and can be understood by a non-technical functional expert

Bottom line:

Using ATL the development efforts were reduced, the code was well organized, it was easier to introduce new additions/changes as the domain model evolved. Also, the functional expert could easily review all mappings and provide his suggestions and confirmation.

References:

- **http://www.eclipse.org/atl/**

- **http://wiki.eclipse.org/ATL/Concepts**