Model Driven Architecture for FHIR

Executive Summary 1

Project Deliverables for FHA 2

UML Representation of FHIR DSTU 2 3

Categories of FHIR model elements 3

Challenges Mapping FHIR Structures to UML Standard 4

MDHT on FHIR 5

Importing FHIR Specifications to UML 5

Viewing and Editing UML FHIR Models 6

Exporting FHIR Specifications from UML 11

Generating a FHIR Profile from FHIM 13

# Executive Summary

A Model Driven Architecture (MDA) separates the business and application logic from the underlying implementation platform technology. The Model Driven Health Tools (MDHT) open source project was created in 2008 to use MDA methodology and development lifecycle for supporting the unique requirements of healthcare clinical information models and interoperability standards. The MDHT team created a set of tooling components that are based on open standards and open source frameworks for MDA. As of August 2015, the MDHT project is part of the modeling tools project at Eclipse.org.

In July 2015, a project was initiated by FHA to extend the MDHT open source platform with support for the HL7 FHIR draft standard, and to enable FHIR to be part of an MDA solution that leverages the Federal Health Information Model (FHIM). A UML model and style guide for the FHIR specification is required before tools can be implemented to map or generate FHIR from the FHIM. Several steps were required to prepare MDHT for use in FHIR design, implementation, and model transformation. The following capabilities were delivered on September 29, 2015:

* Import the HL7 FHIR specification data types, resources, and profiles into a UML model representation. This task required the team to perform a detailed analysis of the FHIR computable “structure definition” format, design an optimal representation in UML, and write a Java application to automatically import the FHIR specification to UML.
* Use the UML model representation for FHIR profiles to generate FHIR Structure Definition files that may be loaded into any FHIR-compliant server.
* Enhance MDHT user interface for viewing and editing FHIR UML models such that it is more accessible to non-developer clinical informaticists.
* Develop a proof of concept application demonstrating how these new MDHT features for FHIR can be used with FHIM models to map, transform, and generate FHIR profiles from the FHIM logical model.

There is a growing body of FHIR profiles being developed by HL7 in projects led by ONC S&I framework initiatives, e.g. Data Access Framework (DAF) and Clinical Quality Framework (CQF). Using UML models created in this project that were imported from the DAF and CQF FHIR profiles, their elements may be mapped to FHIM elements as the *authoritative source* for their semantic definition and use case context. This will be beneficial as a secondary analysis even when the FHIR specifications were not generated from the FHIM models. New future FHIR profile specifications may be developed by transforming from the FHIM logical model to a UML representation for FHIR, and then generating to FHIR standard artifacts, e.g. for applications that support VA/DoD interoperability. All of these FHIR profiles will then have a mapping to common semantic definitions in the FHIM.

# Project Deliverables for FHA

Several steps were required to prepare MDHT for use in FHIR design, implementation, and model transformation. The following capabilities were developed and delivered on September 29, 2015:

* Import the HL7 FHIR specification data types, resources, and profiles into a UML model representation.
  + **Deliverable**: a UML model imported from the HL7 FHIR September 2015 DSTU, including profiles for DAF, QICore, and USLab.
  + **Deliverable**: Eclipse tooling with user interface to select and import FHIR specification files into UML.
* Use the UML model representation for FHIR profiles to implement generation of FHIR StructureDefinition files that may be loaded into any compliant FHIR server.
  + **Deliverable**: Eclipse tooling with user interface to generate a FHIR XML specification file from a UML class that represents a FHIR constraint profile.
* Enhance MDHT user interface for viewing and editing FHIR UML models such that it is more accessible to non-developer clinical informaticists.
  + **Deliverable**: Eclipse tooling with user interface enhancements for FHIR modeling style. Examples screenshots are included in this report.
* Develop a proof of concept demonstrating how these new MDHT features for FHIR can be used with FHIM models.
  + **Deliverable**: QVT (Query/View/Transformation) script generating a FHIR profile from a FHIM Use Case.

The MDHT project team is almost finished migrating its open source project home to Eclipse.org. The new home page is located here:

* https://projects.eclipse.org/projects/modeling.mdht

All of the source code developed as part of this FHA project will be available from the Eclipse repository linked to the MDHT project home page. However, the migration and configuration of MDHT at Eclipse is not yet complete, so this project’s deliverables are available in a temporary open source repository at:

* https://github.com/xmlmodeling/mdht-fhir

# UML Representation of FHIR DSTU 2

The UML modeling style for representing constraint-based standards is an integral part of MDHT methodology and tools and is common to most healthcare interoperability standards, i.e. a base reference model (CDA, FHIR, or NIEM Core) is specialized or profiled for a particular domain by defining constraints on the base model. However, data exchange is still defined by the structure of the base reference model. These model constraints include terminology and value set codes that are required within a particular domain profile, constraints on the allowed data type, or more restrictive cardinality.

The constraint-based representation of FHIR is specified using “profiles”, and each profile is defined in a computable format using a FHIR Structure Definition. Several links for the FHIR DSTU 2, September 2015 version are included here:

* <http://hl7.org/fhir/profiling.html>
* <http://hl7.org/fhir/extensibility.html>
* <http://hl7.org/fhir/structuredefinition.html>
* <http://hl7.org/fhir/valueset.html>

To perform the work completed in this contract task order, all of these FHIR resources (and many other referenced from them) were analyzed in detail to understand their intended use and to parse and process the published FHIR specification distributed using the Structure Definition and Value Set resources in XML format.

## Categories of FHIR model elements

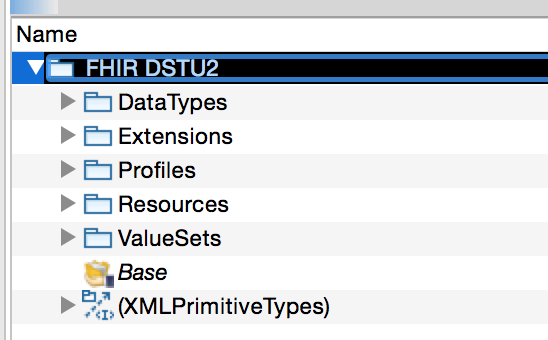
The FHIR core specification is distributed as Structure Definition and Value Set files, in either XML for JSON format. The XML format was used to parse, transform, and import to UML models. The FHIR specification can be divided into these rough categories:

* Data Types
* Resources
* Value Sets
* Extensions
* Profiles

The current HL7 FHIR DSTU 2, September 2015 release also includes several U.S. localized profiles that were developed in collaboration with ONC S&I initiatives. These profiles include:

* Data Access Framework (DAF)
* Clinical Quality Framework (CQF), published the QICore FHIR implementation guide
* US Lab
* Extension elements (patient, pharmacy, etc.)

The complete FHIR DSTU 2 distribution was imported into UML. UML packages were created by the import tool to categorize FHIR artifacts along the lines described above, resulting in a top-level UML package structure as shown in this screenshot:



## Challenges Mapping FHIR Structures to UML Standard

There were a number of challenges when mapping FHIR specification structures into the UML metamodel. We are confident that adequate solutions were developed for UML representation, which includes definition of UML stereotype extensions to the UML metamodel. Brief summaries of several mapping challenges are described below.

Whenever possible, the native UML specification semantics were leveraged and mapped to corresponding semantics in the FHIR structure definitions. Thus, validating the integrity of the UML model as per UML semantics (using any complete UML tool implementation) will also validate many corresponding aspects of FHIR semantics. A few notes about this semantic alignment are included below.

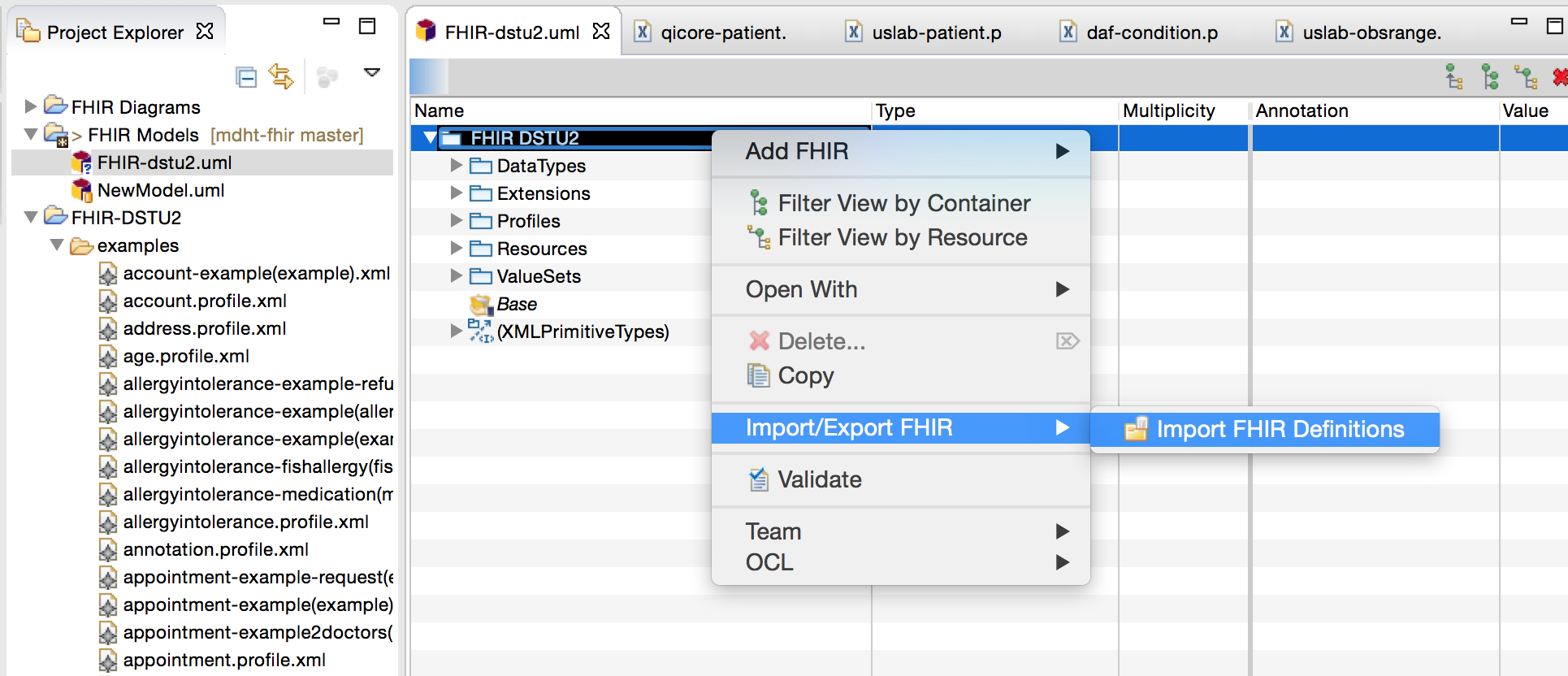
* **Element type choice**
  + In UML, each class attribute has exactly one type, which might be a primitive data type or reference to another class. In FHIR, each structure element may have a choice of types from a list of specified data types or references to other resource types. And referenced types might be required to satisfy a particular profile.
  + The UML property was extended with a <<TypeChoice>> stereotype that includes a list of UML classes, where these UML classes represent the corresponding FHIR data type, core resource, or profiled type.
* **Wildcard elements and naming**
  + Most core FHIR resources include “wildcard” elements with names such as ‘value[x]’ or ‘effective[x]’. This is tied to the element type choice, such that the [x] is replaced by a data type name from that element type choice list. In XML for JSON serialization, the element name has the [x] replaced by the type name. Thus, you get element name such as: valueQuantity, valueCodeableConcept, effectiveDateTime, etc.
  + This use of wildcard element is further complicated by (sometimes inconsistent) use of both ‘value[x]’ and ‘valueCodeableConcept’ (or any other valid combination of wildcard and type name) in element names used in FHIR profile path constraints.
  + This complication in naming has been mostly sorted out in the import/export to UML, but additional testing is needed.
  + Renaming of FHIR wildcard element is mapped to UML property redefinition semantics.
* **Slicing**
  + <http://hl7.org/fhir/profiling.html#slicing>
  + A FHIR structure element may be “sliced”, e.g. to allow for alternative data types, or to enable separate elements for each code in a specified value set. The UML solution is mostly complete and implemented, although some work is still required to create a UML stereotype for slicing “discriminators” and refine the FHIR/UML import/export logic.
  + Sliced elements in FHIR are mapped to UML property subset semantics.
* **Extensions as reusable properties**
  + <https://www.hl7.org/fhir/extensibility.html>
  + Examples of the UML representation for FHIR extension definition and use are included in the next section as part of the QICore-Patient example.
  + At first glace, one would expect FHIR extensions to be like adding a new attribute to a UML subclass. But, in fact, FHIR extension definition and use is just another kind of constraint on a base type.

# MDHT on FHIR

The following sections describe new functionality added to the MDHT platform for importing FHIR specifications to UML, viewing and editing FHIR UML models, and generating FHIR structure definitions from UML.

## Importing FHIR Specifications to UML

A user can download FHIR specification files from HL7, or select StructureDefinition files from other sources. The XML format for FHIR StructureDefinition resource is required. For HL7 specifications, you can download the “validation” archive that includes files containing FHIR Bundles, or download the “examples” archive that contains individual files for each structure definition. In the screenshot shown below, the “examples” folder is shown. Choose a UML package and use the context menu comment to select a folder, and all FHIR data type, resource, and profile definitions are imported to UML.

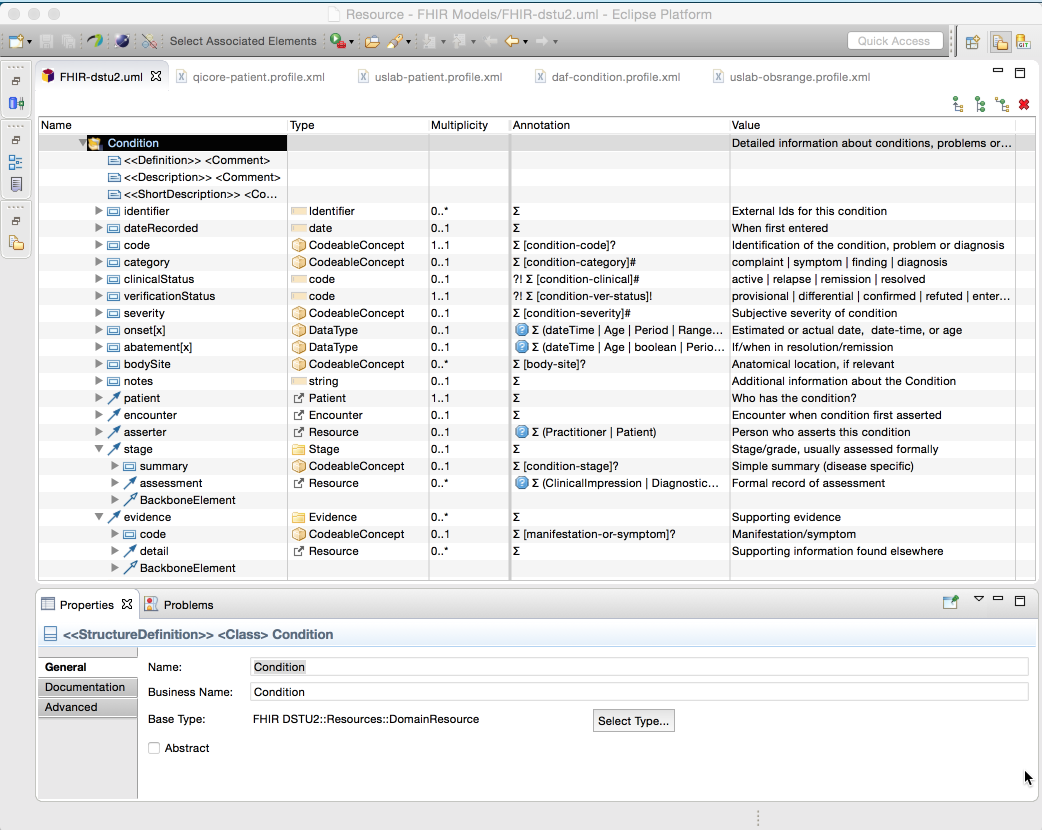


## Viewing and Editing UML FHIR Models

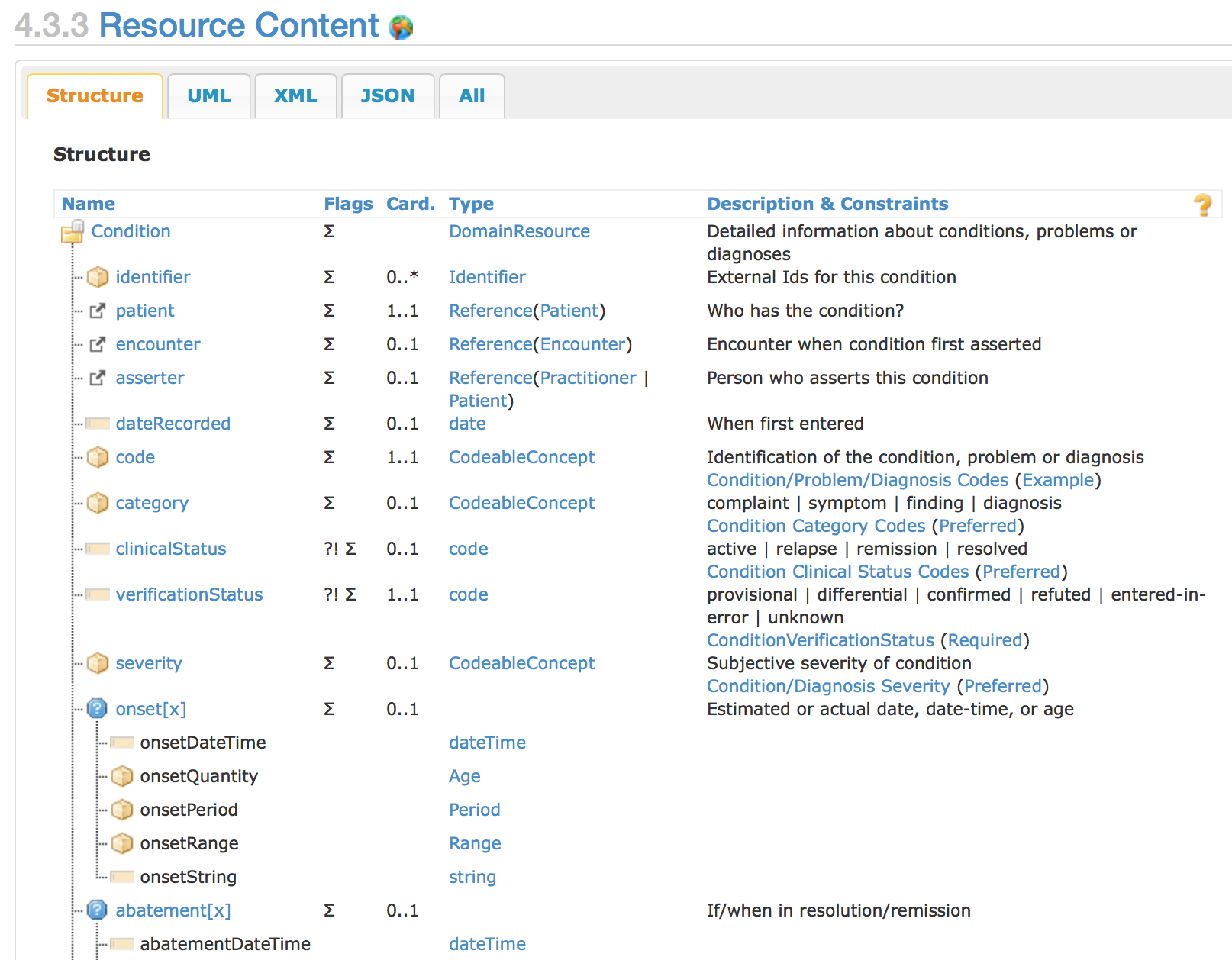
The existing MDHT UML table editor/viewer was customized to include FHIR icons and annotations. The enhancements include:

* Use same icons from the HL7 FHIR specification web pages to indicate type of FHIR element in the UML model.
* Customize the “Annotation” column to show FHIR metadata, e.g. FHIR “flags” and value set binding name.
* Display the FHIR element “short description” in the Value column of UML table editor.

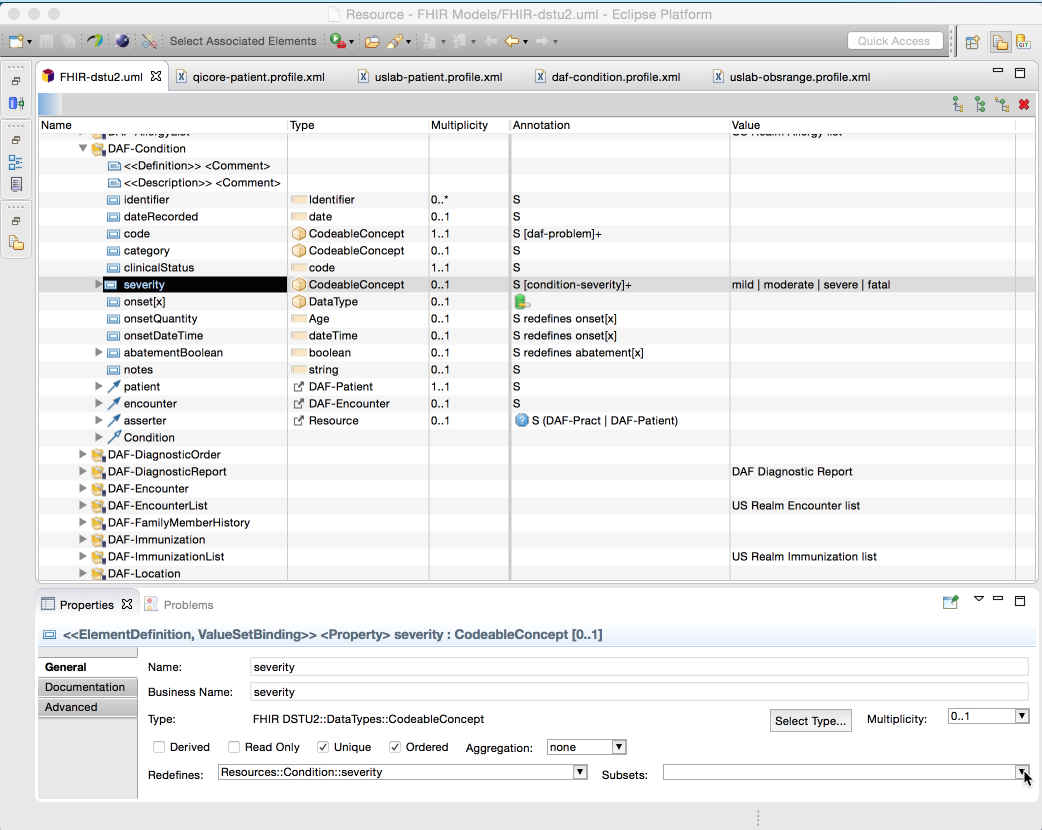
UML model for FHIR Condition resource:



For comparison, the same resource definition displayed on HL7’s web site in HTML appears below. The objective was to provide a similar appearance and structure in UML editing tools that would be familiar to developers working with HL7 published specifications.

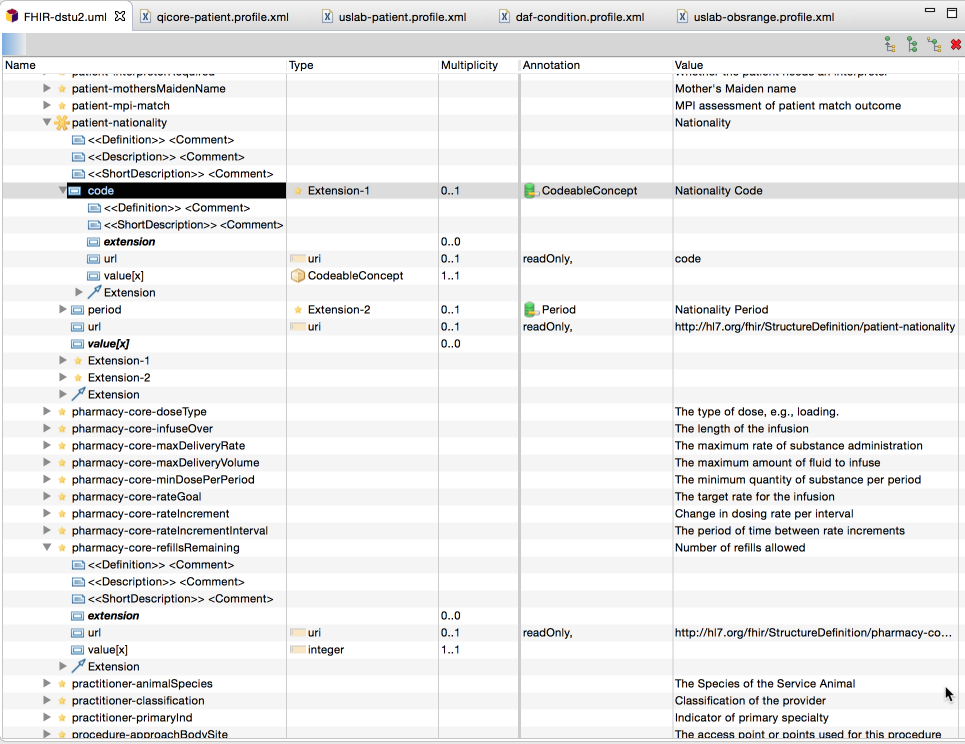


The MDHT tooling also supports importing derived profiles, i.e. profiles that are derived from other profiles as their base. The “differential” constraints and extensions (those different from base profile) are imported into a UML class that is a subclass of the base profile class. In the example screenshot below, the UML class is shown for the DAF-Condition profile that is derived from the FHIR defined resource Condition.

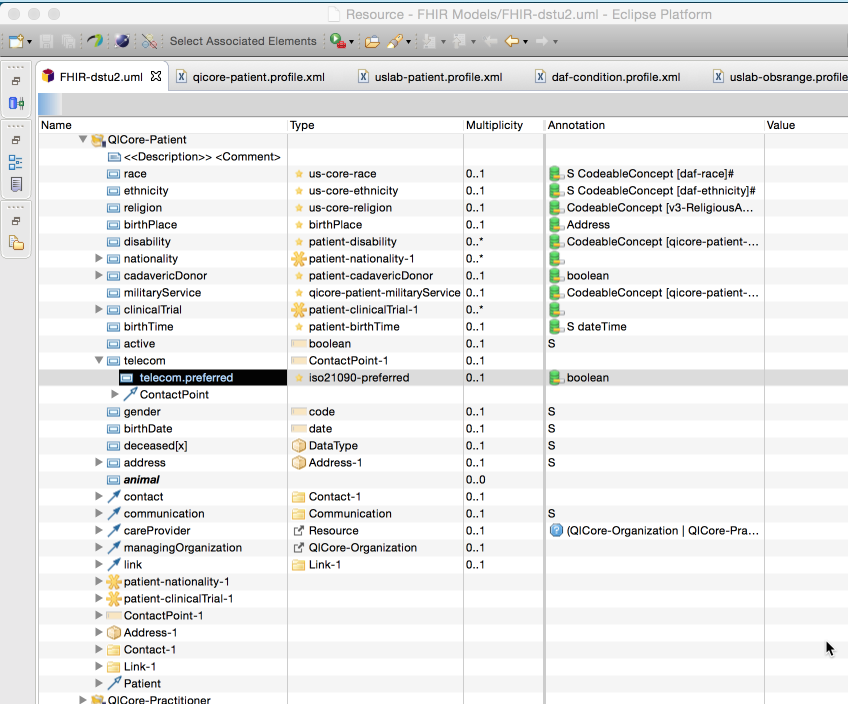


A significant amount of analysis work was performed as part of this contract to design a mapping from FHIR extension definitions to a usable UML representation. A FHIR extension is defined as a constraint on the Extension data type, then that extension may be included in the profile constraints for a resource. Two example screenshots from MDHT are included below.

First, all HL7 defined extensions are imported to UML classes that are subclasses of the Extension data type class. Some FHIR extensions, such as patient-nationality shown below, are “complex extensions”, i.e. extensions that embed other extensions in their content. The patient-nationality includes embedded extensions for ‘code’ and ‘period’.



The following screenshot shows the profile UML class for QICore-Patient, where 10 extensions are added to Patient by “slicing” the inherited extension attribute. This is the most direct mapping from FHIR representation to UML representation. Additional future research will investigate alternative UML representation for profile extensions, e.g. simply adding a new attribute that is not inherited from the base resource. However, an alternative such as this one would require more transformation to create the necessary FHIR slicing constraints when exporting to a FHIR structure definition.

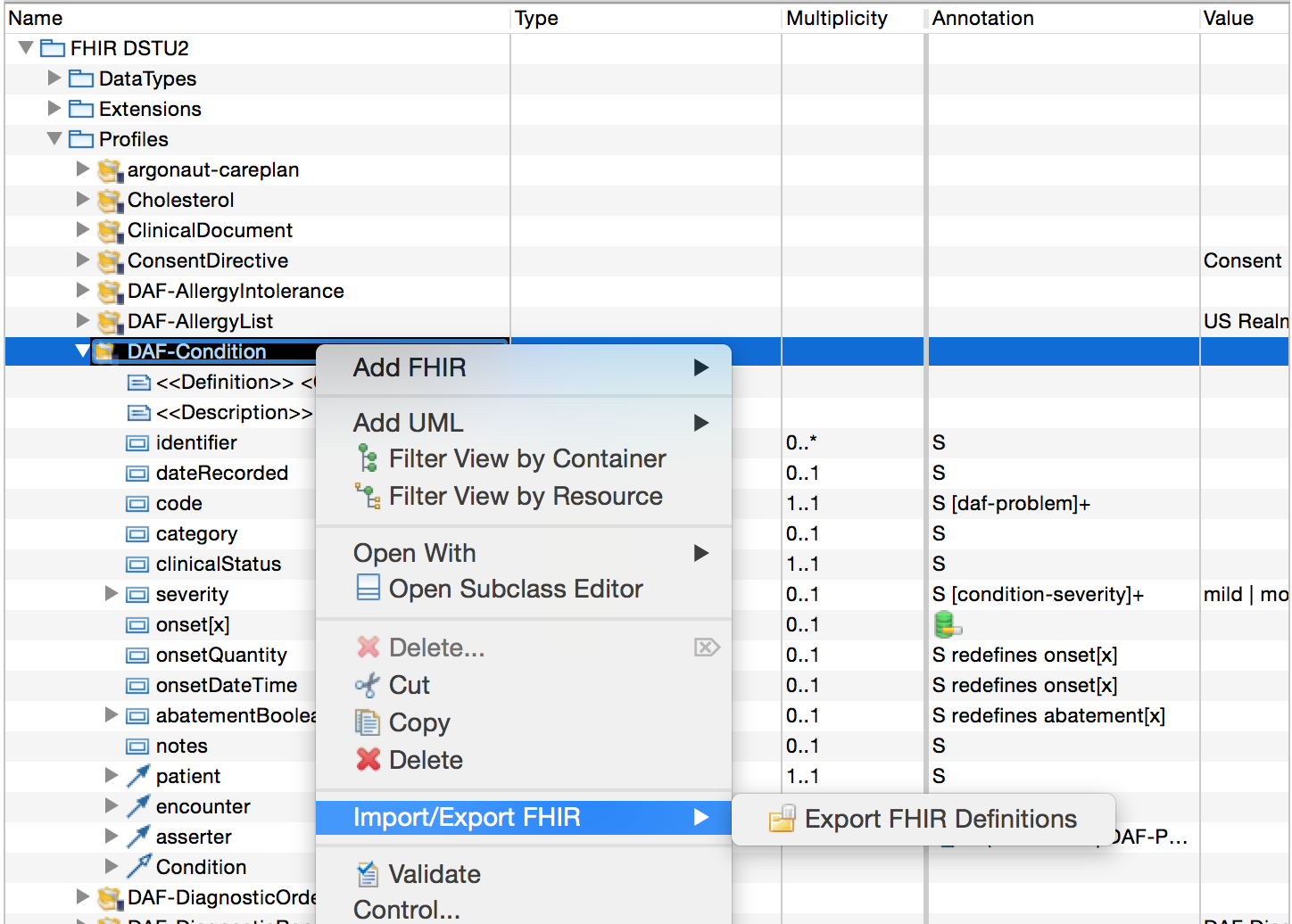


## Exporting FHIR Specifications from UML

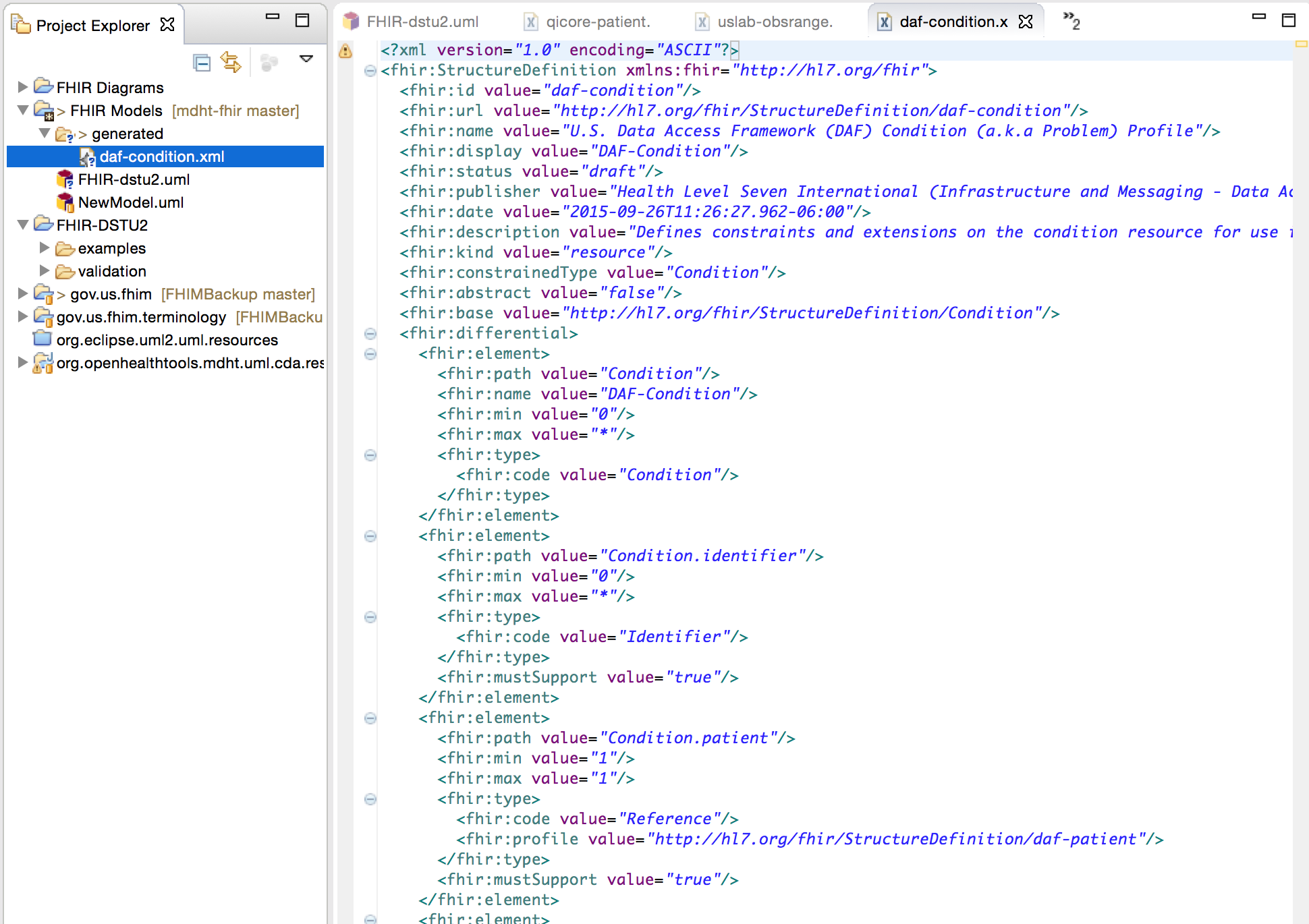
This project’s support for MDA and FHIR must include both importing existing FHIR specifications to UML, and generating FHIR structure definition resources from UML. These generated files can then be posted to FHIR servers and used to support applications. As a demonstration of the capability that was implemented in MDHT, the following screenshots illustrate generating FHIR structure definition files from the DAF-Condition profile. Other scenarios that would use this same capability include:

* Designing new FHIR profiles from within MDHT UML editor, e.g. modifying the DAF profiles, creating new ONC FHIR profiles, VA developers creating FHIR profiles as part of VistA Evolution work, etc.
* Transforming FHIM model content to subclasses of FHIR base resources or subclasses of existing profiles, such as specialization of DAF-Condition. These transform-generated UML profile models would then be generated to FHIR structure files for deployment in applications.

The MDHT table editor includes a context menu command to invoke the export:



The exported FHIR definition file is (currently) added to a “generated” folder beside the model file and opened in an XML editor:



# Generating a FHIR Profile from FHIM

The initial version of the FHIM-to-FHIR Model-to-Model transformation is a QVT script run from an eclipse environment. The Script leverages the contents from the FHIM model to create a FHIR profile. The purpose of the proof of concept is understand the complete content required by the FHIR UML tooling in order to extend/enhance if necessary the current FHIM mapping. The result is a UML model containing the FHIR profile generated from the FHIM.

