This article to appear in *HL7 News*, April 2009.

Modeling CDA with Standards-Based Tools

By Dave Carlson, Project Lead, Open Health Tools (OHT), UML Modeling Project;
Rich Rogers, Co-Chair, HL7 SOA Work Group, and IBM OHT board member;
and Sarah Knoop, John Timm, IBM Healthcare Systems Research

The Clinical Document Architecture (CDA) defines a general document structure for exchanging clinical information. The CDA information model is specified as a restriction of the HL7's Reference Information Model (RIM), although most designers and users work with the CDA XML schemas. CDA applications are often based on an *implementation guide* that defines a standard document structure using *templates*, usually contained within CDA document sections. One of the most widely used CDA implementation guides is the Continuity of Care Document (CCD) which is composed of more than 50 templates.

A new HL7 project has been proposed within Structured Documents that will develop the approach and tooling requirements for specifying CDA templates and implementation guides using the Unified Modeling Language (UML). UML is a widely adopted standard for modeling software systems (www.uml.org). There is a thriving ecosystem of users, educational resources, and open source and commercial tools that support UML. The UML is often used in combination with the Object Constraint Language (OCL) that is capable of specifying semantic constraints such as those used in CDA templates.

This HL7 project will provide the requirements and analysis for new tools developed within an Open Health Tools (OHT) project, Modeling Tools for Healthcare (modeling-mdt.projects.openhealthtools.org). Open Health Tools is an organization with a mission to provide open source software for healthcare interoperability. HL7 is a member of the board of Open Health Tools.

This OHT modeling tools project was formed in April 2008 to build healthcare tools that are based on industry standard modeling languages, such as UML, OCL, and the Web Ontology Language (OWL). The tools include UML profile extensions with stereotypes for HL7 metadata and modeling style, plus XML schema generation for HL7 message models. We have formed a new OHT modeling subproject that is dedicated to supporting the creation and implementation of CDA implementation guides using UML models. Working together, we anticipate that three distinct groups in the user community will benefit from our project's tools: template authors, instance creators, and tool and application developers.

New CDA templates and implementation guides are authored by the HL7 Structured Documents Working Group, and may be further constrained by the Health Information Technology Standards Panel (HITSP) in the US Realm (e.g. C 32) and Integrating the Healthcare Enterprise (IHE), and by other organizations that need a well-defined clinical document structure for exchanging specific clinical content. It must be possible to validate that the templates are correctly defined as constraints on the CDA model or on a higher-level template. For example, the HITSP C 32 guide is a restriction of the CCD, which is a restriction of the CDA, which is a restriction of the RIM.

A second group of tool users are those who create high-quality CDA document instances. All CDA instances are XML documents that are governed by the general CDA schemas. However, most instances also must be valid with respect to all template

constraints specified in the implementation guide that describes each class of instances. CDA modeling tools may be used in the development of graphical editors that are specialized for a particular implementation guide, such as the CCD. Or instances may be created by application developers as part of exporting clinical information from electronic health record systems. We expect that our modeling tools will enable generation of robust application programming utilities for automated processing of CDA instances.

The third group of tool users are developers of applications and other tools. Following a model-driven development approach, the UML models created as part of the CDA template design process will be used to generate Java APIs for processing and validating document instances that conform to those templates. Other future generated runtime tools may include XSLT transforms, Schematron rules, and XForms support for Web-based editors. These runtime platforms may be used to create service-oriented applications, instance editor tools, database extract and load adaptors, and other implementations specialized to the modeled implementation guide.

The overall goal of these coordinated HL7 and Open Health Tools projects is to accelerate the creation, verification, publication, and deployment of CDA implementation guides. We hope to make CDA more accessible by enabling a broader community of users to create high quality guides, ballot the guides through HL7, and deploy approved guides in production systems. We are starting by creating a specification of CDA, CCD, and C 32 in UML. We welcome your participation in these new projects!