CDAR2_IG_GREENMOD4CCD_R1_INFORM_2011APR

HL7 Implementation Guide for CDA® Release 2: greenCDA Modules for CCD, Release 1 (US Realm)

April, 2011

HL7 Informative Document

Sponsored by:
Structured Documents Work Group

IMPORTANT NOTES:

HL7 licenses its standards and select IP free of charge. **If you did not acquire a free license from HL7 for this document,** you are not authorized to access or make any use of it. To obtain a free license, please visit http://www.HL7.org/implement/standards/index.cfm.

If you are the individual that obtained the license for this HL7 Standard, specification or other freely licensed work (in each and every instance "Specified Material"), the following describes the permitted uses of the Material.

A. HL7 INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS, who register and agree to the terms of HL7's license, are authorized, without additional charge, to read, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part without paying license fees to HL7.

INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS wishing to incorporate additional items of Special Material in whole or part, into products and services, or to enjoy additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS as noted below, must become ORGANIZATIONAL MEMBERS of HL7.

B. HL7 ORGANIZATION MEMBERS, who register and agree to the terms of HL7's License, are authorized, without additional charge, on a perpetual (except as provided for in the full license terms governing the Material), non-exclusive and worldwide basis, the right to (a) download, copy (for internal purposes only) and share this Material with your employees and consultants for study purposes, and (b) utilize the Material for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, Compliant Products, in all cases subject to the conditions set forth in this Agreement and any relevant patent and other intellectual property rights of third parties (which may include members of HL7). No other license, sublicense, or other rights of any kind are granted under this Agreement.

C. NON-MEMBERS, who register and agree to the terms of HL7's IP policy for Specified Material, are authorized, without additional charge, to read and use the Specified Material for evaluating whether to implement, or in implementing, the Specified Material, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part.

NON-MEMBERS wishing to incorporate additional items of Specified Material in whole or part, into products and services, or to enjoy the additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS, as noted above, must become ORGANIZATIONAL MEMBERS of HL7.

Please see http://www.HL7.org/legal/ippolicy.cfm for the full license terms governing the Material.

Co-Chair/Co-Editor	Liora Alschuler Alschuler Associates, LLC liora@alschulerassociates.com
Co-Chair/Co-Editor	Robert H. Dolin, MD Semantically Yours, LLC bobdolin@gmail.com
Co-Chair	Calvin Beebe Mayo Clinic cbeebe@mayo.edu
Co-Chair	Grahame Grieve Health Intersections Pty Ltd grahame@healthintersections.com.au
Co-Chair	Austin Kreisler SAIC - Science Applications International Corporation austin.j.kreisler@saic.com
Co-Editor:	Kate Hamilton Alschuler Associates, LLC kate@alschulerassociates.com
Co-Editor:	Sean McIlvenna Alschuler Associates, LLC sean@alschulerassociates.com
Co-Editor:	Lauren Wood Alschuler Associates, LLC lauren@textuality.com
Co-Editor:	Rick Geimer Alschuler Associates, LLC rick@alschulerassociates.com
Current Working Group also includes:	Gerald Beuchelt, Keith Boone (as former Co-Chair during the first ballot), Andrew J.Gregorowicz

Acknowledgments

This document was produced and developed through the volunteer efforts of Alschuler Associates, LLC, and Semantically Yours, LLC, in cooperation with and under the auspices of the Health Level Seven (HL7) Structured Documents Work Group.

Our experience so far is that every person with an interest in the topics of easier or simpler CDA has at least a few concrete ideas and assumptions about what "easier" and "simpler" mean and how to achieve these goals. For good discussion, it is important to make the assumptions explicit. This document is intended to stimulate discussion, further experimentation, and full development of a simplified approach to CDA implementation.

Revision History

Rev	Date	By Whom	Changes
1.0	27 August 2010	Rick Geimer	Initial ballot document
2.0	3 March 2011	Rick Geimer	First release

Table of Contents

Table of Figures

Table of Tables

1 INTRODUCTION

The Clinical Document Architecture Release 2.0 (CDA R2) addresses universal requirements for exchange and management of structured clinical documents. The approach described in this document, greenCDA, maintains the full utility of the CDA R2 while defining a method of working with an implementation-specific XML (Extensible Markup Language) that is easier to implement.

greenCDA, if adopted, will provide a strategy for developers who generate templated CDA. This is not a substitute for CDA.

The approach described here simplifies instance creation while it asserts as a primary principle that any simplification must also define a method to deliver valid, normative CDA.

This approach is called “greenCDA” because it is good for the environment!

1.1 Contents of the Ballot Package

The following files comprise the ballot:

Table 1: Ballot Contents

Filename	Description	Status
CDAR2_IG_GREENMOD4CCD.doc	Implementation Methodology in Microsoft Word 2003 format.	Balloted
CDAR2_IG_GREENMOD4CCD.pdf	Implementation Methodology in PDF format.	Balloted
green_c32_full.xml	A sample XML instance conforming to green_c32.xsd	Example
green_ccd.xslt	An XSLT 2.0 transform to convert green_ccd.xml to normative CDA XML.	Example
narrative.xslt	A reusable XSLT 1.0 transform from XHTML to CDA narrative markup (does not handle the XHTML <img/> tag	Example
green_c32_full_normative_cda.xml	The result of running green_ccd.xslt against green_ccd.xml	Example
green_c32.xsd	Sample greenCDA schema for C32	Example
green_c32_base.xsd	Sample base module for C32. Defines some reusable participants such as author and information source.	Example
green_c32_datatypes.xsd	Sample greenCDA datatypes schema	Example
green_c32_advance_directives.xsd	Sample greenCDA advance directives schema module	Example
green_c32_conditions.xsd	Sample greenCDA conditions schema module	Example
green_c32_encounters.xsd	Sample greenCDA encounters schema module	Example
green_c32_immunizations.xsd	Sample greenCDA immunizations schema module	Example
green_c32_insurance_providers.xsd	Sample greenCDA insurance providers schema module	Example
green_c32_medications.xsd	Sample greenCDA medications schema module	Example
green_c32_plan_of_care.xsd	Sample greenCDA plan of care schema module	Example
green_c32_pregnancies.xsd	Sample greenCDA pregnancies schema module	Example
green_c32_procedures.xsd	Sample greenCDA procedures schema module	Example
green_c32_results.xsd	Sample greenCDA results schema module	Example
green_c32_vital_signs.xsd	Sample greenCDA vital signs schema module	Example
green_cda_narrative.xsd	An XHTML subset for the greenCDA narrative block.	Example
requirements.xls	An example requirements spreadsheet	Example

1.2 What it Means to be Green and Identifying our Target Audience

This document adopts and documents specific premises about who is to benefit and what it means to be green.

- A greenCDA schema willhave a shorter learning curve and yet deliver valid CDA documents
- be easier to work with for instance generation

The beneficiaries—the people who get the shorter learning curve and the easier model to work with—are developers who must generate CDA documents and who do not know CDA or the Reference Information Model (RIM).

Please keep in mind that the target audience are people who *don't* know CDA or the RIM. People who do know CDA and the Continuity of Care Document (CCD) will likely be the first to evaluate this effort and to them, this approach may seem unusual at first blush.

1.3 Process Flow

Figure 1 shows the design process, which combines implementation-specific requirements with the normative CDA schema and a repository of CDA templates. The results are implementation-specific (green) instances, samples, transforms, and schemas. Note that this is the same process used today to create implementation-specific Schematron rule sets. This green process combines the implementation-specific *.sch rules with a simplified CDA XML Schema document (also known as an XSD).

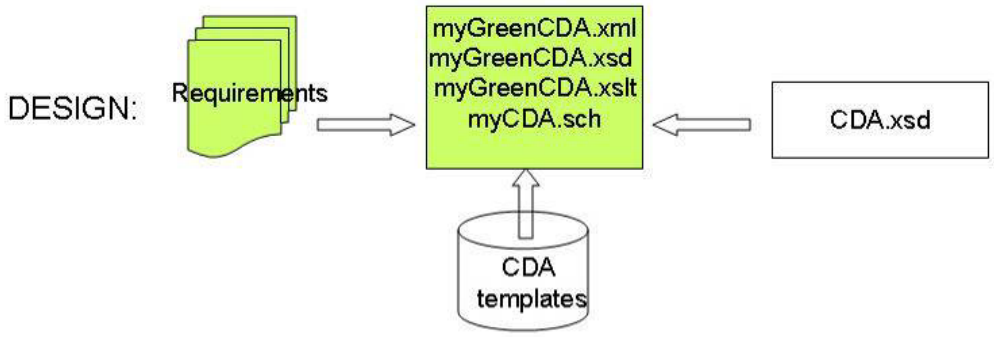

Figure 1: Design process for greenCDA

Figure 2 shows how the green artifacts work locally to produce a normative CDA document (myCDA.xml) that is conformant with CDA R2 and can be validated against the implementation-specific rules (CDA.xsd and myCDA.sch) as is commonly done today.

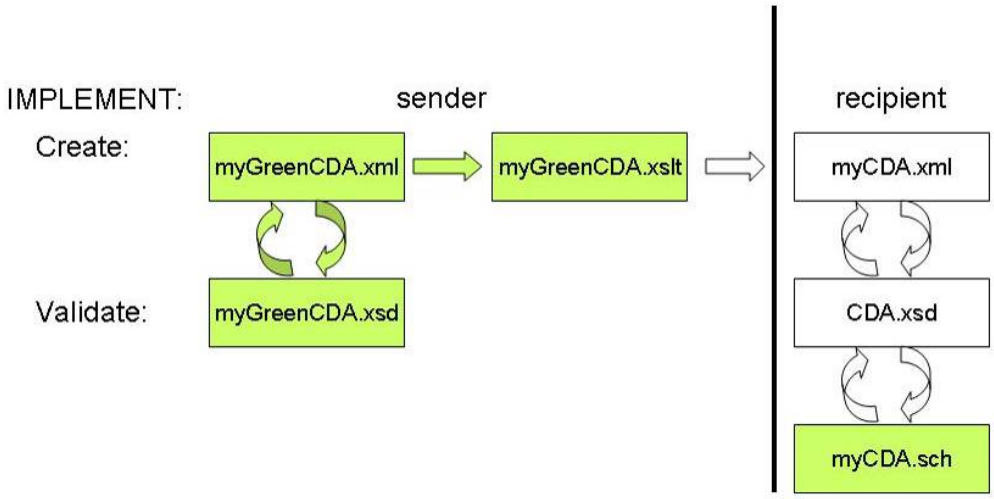

Figure 2: Producing a normative CDA document from greenCDA

2 METHOD FOR DESIGNING GREENCDA IMPLEMENTATIONS

We define a greenCDA implementation as the complete set of schemas, transforms, and other artifacts that are necessary to create a greenCDA instance that can be transformed into the matching full CDA instance.

This greenCDA implementation will be composed of one or more greenCDA modules. A module as defined in this document is a chunk of XML syntax, defined by a schema, and potentially reusable in other schemas via xsi:include. . Examples would be a medications module or a lab test results module, each of which encapsulates the information that would be transmitted in the appropriate section of a complete CDA document. It is not mandatory for a greenCDA implementation to be composed of multiple reusable modules…a single monolithic schema is acceptable.

The following design process discussion uses greenCDA modules as examples, with the understanding that these modules can be combined into greenCDA implementations. Also, this discussion is intended for greenCDA designers; implementers need only use the schemas and transforms provided by designers in order to create CDA documents according to this methodology.

2.1 Creating a "Green" Implementation

A greenCDA implementation must follow the following rules in order to be considered "green".

CONF-GREENCDA-1: A greenCDA implementation SHALL define an intermediate XML form, defined at least by one or more XML Schema modules, and a corresponding transform that is used to convert the intermediate form to a CDA document.

An Example XSLT 2.0 transform from the greenCDA implementation of C32 to fully normative CDA markup is included with this package.

CONF-GREENCDA-2: The XML markup of a greenCDA schema module shall be in a namespace whose URI begins with "urn:hl7-org:greencda".

The provided C32 implementation uses urn:hl7-org:greencda:c32 as the namespace URI.

2.2 Data Elements

The process begins with the definition of the data elements in CDA that should be part of the module. For each element, these data element definitions include the business names and cardinality as well as the mapping to full CDA, including all required underlying CDA constructs. The following example in Figure 3 is from Health Information Technology Standards Panel (HITSP) C32/C83.

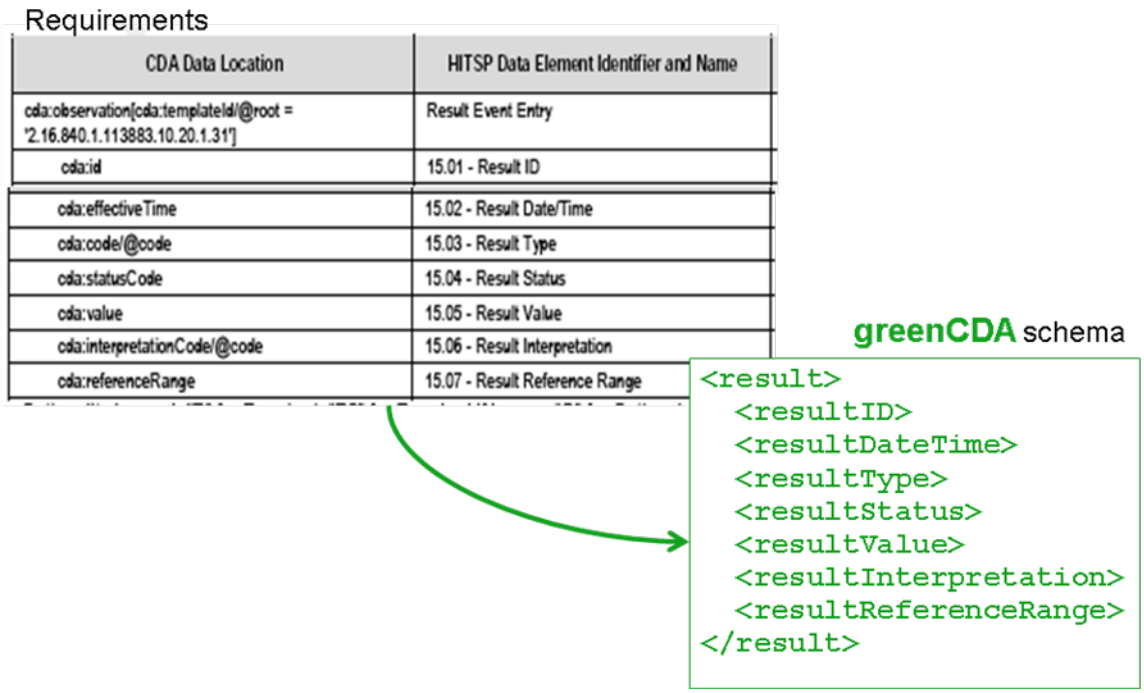

Figure 3: Defining data elements in a module

Note that the above example is only for results, and would not be appropriate for all types of clinical statements. For example "Result status" might be clear to represent that it is the status of the observation, but one could easily see where a "ProblemStatus" would generate confusion because in this case it mechanically transforms to <statusCode> which would be incorrect, (the observation is complete even if the problem is inactive).

Once we have the complete set of data element definitions, we create a simple XML structure that is based on the business names. For simplicity, we have tried to ensure that it is possible to process greenCDA documents using only XML 1.0 + namespaces. Where we had a choice between using elements or attributes, we have opted to use elements. We use nesting primarily when constructs are repeatable or for complex datatypes. Where possible, we avoid using complex types when basic element definitions are sufficient. These restrictions facilitate the use of XML data-binding tools, which allow automatic generation of processing code in many languages.

2.3 Datatypes

We simplify datatypes in a few ways to increase the ability to automate the system. We prefer simple datatypes over complex datatypes. We avoid xsi:type by defining concrete datatypes (early binding) and we use choice constructs where multiple datatypes are allowed. For example, we use actual elements named **`physicalQuantity`** or **`integer`** rather than a generic “value” attribute. As shown in Figure 4, the transform will map the greenCDA datatypes to normative CDA markup with xsi:type populated.

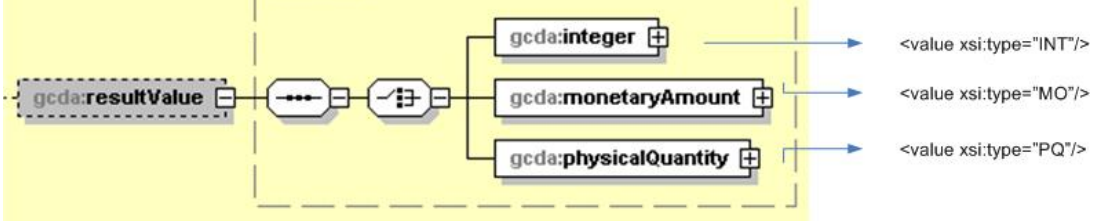

Figure 4: Transforming greenCDA datatypes to normative CDA markup

We are balancing these simplification needs with the goal of staying close to the base CDA types. However, depending on the scenario, different levels of simplification may be appropriate. What has been demonstrated here with CCD is just one example and is not intended to be used as the general method.

This methodology does not remove the requirement to understand HL7 datatypes in general. For example, an implementer would generally need to record date/time values in yyyymmddhhmmss format.

What was done with the data types here with the CCD is just one example. It is not intended to be a general method.

2.4 Terminology

The simplification we propose is to record static value sets in the schema where possible. This reduces (but does not entirely eliminate) the need for secondary validation, as dynamic value sets still require Schematron or a terminology lookup service, or both.

Static value sets consisting of a single value can effectively be eliminated from the schema entirely and just provided by the transform.

2.5 Narrative Text

The C32 implementation provided with this document auto-generates the narrative block from the coded entries. However, not all greenCDA implementations need to take this approach. This section provides guidance for creating a simplified human readable narrative block.

2.5.1 XHTML Markup

The recommended markup for greenCDA narrative text is a subset of XHTML that is readily transformable to CDA narrative block markup. XHTML was chosen because a) many developers are already familiar with it so there is no need to learn new markup, and b) it makes it relatively easy to refactor code that displays documents in a web browser for greenCDA support. Use of XHTML for narrative blocks is a common practice and to the extent that this exists we provide a solution. We recommend limiting the XHTML markup to the tags outlined in Table 2 if implementers wish to reuse the XHTML transformation code provided with this implementation methodology.

Table 2: Recommended XHTML Tags

XHTML	CDA equivalent	Notes
<p>	<paragraph>	
<b>	<content styleCode=”Bold”>	
<i>	<content styleCode=”Italics”>	
<u>	<content styleCode=”Underline”>	
	<list styleCode="ordered">	
	<list styleCode="unordered">	
	<item>	
<a>	<linkHtml>	
<img>	<renderMultiMedia>, <observationMedia>	The transform will need to create an observationMedia entry, then reference that via renderMultiMedia
<table>	<table>	Since CDA already uses the XHTML table model, table and all its sub elements can be easily transformed to CDA markup.

Note: the example narrative block schema and transform provided with this document do not currently handle the <img> tag.

2.5.2 Optionality of the Narrative Block

All CDA documents require a human-readable narrative block. Documents that conform to a greenCDA methodology have a slightly different requirement: The CDA that is generated when the greenCDA document is transformed must have a complete human-readable narrative block that conforms to all the requirements of CDA.

In practice, this means that a narrative block may be omitted from sections in greenCDA documents and schemas, provided that the following conditions are true:

All data in a section is fully coded (i.e., Level 3 data in CDA terms).

The transform that generates the normative CDA markup must have narrative generation capabilities.

The resulting CDA document must have a human-readable narrative and all the entries must have the derived relationship asserted (entry/@typeCode=”DRIV”).

However, it is expected that many, if not most, greenCDA documents will still require a narrative block since narrative content is essential to most clinical documents. If a manually created human-readable narrative is present, the system generating the greenCDA document must ensure that the resulting CDA after transform contains a full narrative block conforming to the rules of CDA (i.e., it must contain all relevant information from the coded entries as well).

This informative document does not address the difficult situation of minimal narrative mixed with coded entries where the manual narrative is preserved and narrative for coded entries is generated for future work. It is likely possible to handle such a situation by interleaving “mini narrative blocks” with coded data.

2.6 Generic Constraints on Schemas and Transforms

There are also generic constraints on the design. The major constraint is that all greenCDA modules must be transformable to full CDA. This means that it must be possible to transform any document instance written according to a greenCDA module schema to a document instance that is a conformant CDA document (usually, this means that it at least validates according to the CDA schema once all extensions have been removed).

Note that another simplification is to use a restricted CDA schema which is only a subset of the full schema if, for example, the module deals with medications only rather than being a full CDA implementation. This is particularly useful when stripping out some unrelated complexity in the data types and vocabulary schema.

Most transforms should be possible with an Extensible Stylesheet Language Transform (XSLT). It may also be possible to design the mapping between full CDA and greenCDA schemas using mapping tools, but this could be difficult for greenCDA documents that contain a lot of narrative markup using mixed content (a combination of plain text and XML markup in an arbitrary order). The greenCDA document instance must contain sufficient data to populate all required CDA elements in the generated CDA document instance, again depending on whether it is a module (i.e., a subset) or a full implementation. However, since some data may be implied by the constraints and therefore supplied in the transform, not all data is required to be physically present in the greenCDA instance.

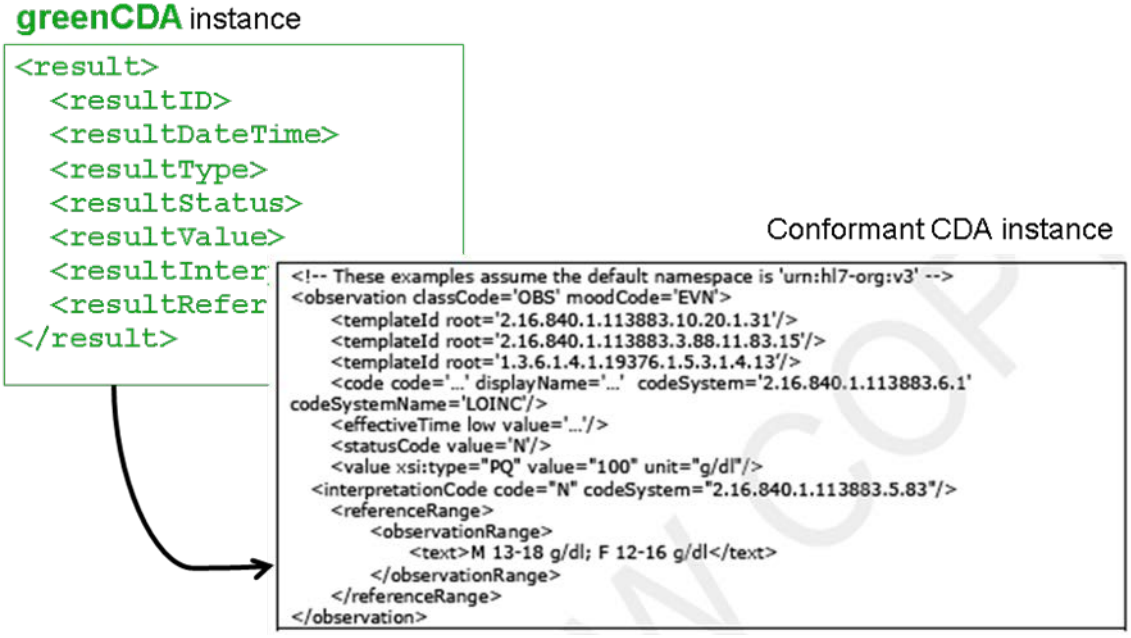

Figure 5: Transforming greenCDA to a conformant CDA instance

The sample transform is a uni-directional transform and is only intended to demonstrate how to generate a normative CDA instance from a greenCDA instance.

2.7 Summary

This methodology starts with a well-defined subset of CDA in each case and defines a matching greenCDA module or implementation (multiple modules combined). We have chosen not to try to define a generic solution for any CDA implementation, but rather to concentrate on those items that are well defined and therefore should be amenable to automation.

3 MODULE REQUIREMENTS DEFINITION

Before designing a greenCDA schema and associated transform for a particular module, it is necessary to define some requirements up front. At a minimum, the following must be defined:

Business names (maps to greenCDA element names)

Cardinality (optional, required, and/or repeatable)

CDA mappings (for transformation)

Table 2 shows the sample module requirements captured for the CCD/C32 Plan of Care section.

Table 3: Sample Requirements Table

Business Name	Cardinality	Data Type	CDA Mapping	Value Set	Notes
resultsOrganizer	1..*		component/structuredBody/component/section/entry/organizer/…		
code	1..1	cd	code		
effectiveTime	1..1	ivl_ts	effectiveTime		
result	1..*		component/entry/observation/…		
resultId	1..*	ii	id		
resultDateTime	1..1	ivl_ts	effectiveTime		
resultType	1..1	cd	code		
resultStatus	1..1	cs	statusCode		
resultValue	1..1	value	value		
resultInterpretation	0..1	cd	interpretationCode		
resultReferenceRange	0..*		referenceRange		
comment	0..*	comment	entryRelationship		

Table 3 is an example of a table that may be used in the design of the greenCDA schema modules to associate business names to normative CDA. The specific format of the requirements table (or the choice of other formats besides a table) is left to the greenCDA schema module designer.

A partial module requirements spreadsheet for C32 is provided with this package as an example. It includes a list of datatypes, the high level document organization, and 3 example sections (results, encounters, and medications). The business names from C32

were used with minimal modifications (removing spaces, etc.), even in cases where we felt better names may be appropriate.

4 OUTSTANDING ISSUES

4.1 Relationship to other HL7 Projects

- What is the relationship between greenCDA and MicroITS?
- How does greenCDA relate to the Neutral Mapping project?

4.2 For consideration during the Trial Use Period

- Is it useful to automate the generation of greenCDA schemas from a formal requirements definition?
- Prototyping a narrative generation strategy that allows for a mix of manually entered narrative and coded data that needs narrative generation in the same section.
- How to deal with co-occurance constraints and other expressional dependencies (i.e. C32 defines the following constraint: "Result Value SHALL be present when the observation/@moodCode is EVN or GOL, and SHALL NOT be present when observation/@moodCode is INT or PRP." The C32 implementation provided with this specification assumes that the moodCode attribute is always EVN, and provides no way to modify this attribute, assuming that EVN is the most common value for moodCode)
- How far should simplification go? (i.e. The example makes no attempt to simplify the use of the HL7 negationInd attribute, even though it is very confusing for implementers. Future implementations may reconsider this approach if deemed necessary.)
- What are the risks associated with using a matching greenCDA process in reverse for processing a CDA document?
- When would it be appropriate to exchange and/or store greenCDA instances rather than full CDA documents

5 REFERENCES

- CCD: Continuity of Care Document: American Society for Testing and Materials (ASTM)/HL7
- LOINC®: Logical Observation Identifiers Names and Codes, Regenstrief Institute, Inc. (http://loinc.org)
- SNOMED CT®: SNOMED Clinical Terms, 2002, SNOMED International Organization (http://www.ihtsdo.org/snomed-ct)

APPENDIX A — ACRONYMS AND ABBREVIATIONS

ASTM	American Society for Testing and Materials
CCD	Continuity of Care Document
CDA	Clinical Document Architecture
HITSP	Health Information Technology Standards Panel
HL7	Health Level Seven
IHTSDO	International Health Terminology Standard Development Organisation
LOINC	Logical Observation Identifiers Names and Codes
R2	Release 2
RIM	Reference Information Model
SNOMED CT	SNOMED Clinical Terms
XML	Extensible Markup Language
XSD	XML Schema Definition