HL7 Vocabularies: Bridging Information Models and Vocabulary in HL7

October 26, 2005
Alexandria, Virginia

Woody Beeler
Russell Hamm

http://www.hl7.org
Outline of today’s presentation

• Terminology & Information Models – foundations of HL7 standard information structures
• HL7 Modeling & Methodology Technical Committee
  – Coded terminologies & HL7 Standard Information Structures
• HL7 Vocabulary Technical Committee
  – The HL7 Terminology Model
  – Version 3 coded data types (with XML ITS examples)
  – Vocabulary Binding to Information Models
  – Common Terminology Services (CTS) specification
Why Standard Information Structures

• One of the fundamental goals of computerized medical information is that of precise, accurate and unambiguous communication.

• This communication is required
  – amongst healthcare practitioners,
  – between healthcare practitioners and their patients,
  – between healthcare practitioners and a wide variety of financial, statistical, research and regulatory agencies.
Any meaningful exchange of utterances depends upon the prior existence of an agreed upon set of semantic and syntactic rules.

Basis for Communication (2)

- Information models define the basic relationships between elements of an information structure, whether message or document.
- HL7-defined terminologies allow specialization (refinement) of the information models to serve a particular purpose.
- Standard HL7-designated terminologies provide the fundamental values to be used with encoded data (over 45% of health care data).
- In HL7, the Reference Information Model and the Vocabulary combine to provide **the rules and semantics** for expressing health care information.
The HL7 Modeling & Methodology Technical Committee
Committee Mission

Modeling & Methodology TC:
Responsible for creating and maintaining the HL7 development methodology, and maintaining a Reference Model that reflects the shared models that are developed and used by the HL7 Functional Committees.
Modeling & Methodology Activities

• Ongoing projects:
  – Define and Maintain “HL7 Development Framework” (HDF) – a development methodology to be used by HL7 in defining messages and other information structures for healthcare systems interoperability
  – Define and Maintain a single Reference Information Model (RIM) for use by HL7 developers, based upon an open, consensus-driven harmonization process
Terminology & HL7 Information Structures
-- Definition and Refinement
HL7 Reference Information Model (RIM)
Subject Area: a major partition of an information model.

Class: something about which information is collected.

Relationship: an affiliation between two classes.

Attribute: information about a class.

Data Type: a specification of the format of an attribute.
4 Primary Subject Areas
• 35 Classes
• 181 Attributes
• 9 Associations

… physical things of interest
• Persons
• Organizations,
• 28 others

… take on Roles in health care
• Patient, Physician, Employer

… participate in Acts
• Encounters, Observations, procedures, medications

Entities in Roles
participate as …
• Performer, Author, target
**Action** – the focus of health care messaging

- The reason we want to automate health care data is to be able to document the *actions* taken to treat a patient:
  - A request or order for a test is an *action*
  - The report of the test result is an *action*
  - Creating a diagnosis based on test results is an *action*
  - Prescribing treatment based on the diagnosis is an *action*

- In simple terms, a medical record is a record of each of the individual *actions* that make up the diagnosis, treatment and care of a patient.
Five core concepts of the RIM

- Every happening is an **Act**
  - Procedures, observations, medications, supply, registration, etc.

- Acts are related through an **ActRelationship**
  - composition, preconditions, revisions, support, etc.

- **Participation** defines the context for an Act
  - author, performer, subject, location, etc.

- The participants are **Roles**
  - patient, provider, practitioner, specimen, employee etc.

- Roles are played by **Entities**
  - persons, organizations, material, places, devices, etc.
RIM Core Classes

Entity
- Organization
- Living Subject
- Person
- Material
- Place

Role
- Patient
- Employee
- Licensed Entity
- Access

Participation
- 0..*
- Link 1

Role Link
- 0..*
- 0..*
- 1

Act
- Relationship
- Procedure
- Observation
- Patient Enc’nt’r
- Substance Adm
- Supply
- Referral
- Financial act
- Working list
- Account

Persistent concept
Phrase
Classes may have coded attributes

Class: Act

Description: A type of stakeholder. An individual human being.

Associations for: Act

- **target** :: (1..1) **Act** :: **inboundRelationship** :: (0..*)
- **source** :: (1..1) **Act** :: **outboundRelationship** :: (0..*)

Attributes of: Act

- **classCode** : CNE {ActClass}
- **moodCode** : CNE {ActMood}
- **code** : CWE {ActCode}
RIM – Abstract model with encoded detail

Entity

Role

Participation

Activities

- 9 class-subtypes
- 27 enumerated subtypes
- 35 Classes
- 181 Attributes
- 82 “Coded” attributes (45 %)

- 4 class-subtypes
- 75 enumerated subtypes

- 15 class-subtypes
- 49 enumerated subtypes
- 1 class-subtype
- 50 enumerated subtypes

- 0 class-subtypes
- 56 enumerated subtypes

4 class-subtypes

75 enumerated subtypes

15 class-subtypes

49 enumerated subtypes

1 class-subtype

50 enumerated subtypes

0 class-subtypes

56 enumerated subtypes
Is “Act” sufficient?

• How can a single act class represent all of the elements of clinical action – their definition, request, order, report?

• Answer: Terminology (codes) that control
  – the “sub-type” of Act (disjoint, incomplete), and
  – The “mood” of the action, in the sense of the mood of a verb in grammar

• “A code specifying whether the Act is an activity that has happened, can happen, is happening, is intended to happen, or is requested/demanded to happen. “
“Mood” of a Verb (or action)?

“Yes, verbs have moods, but these ‘moods’ have nothing to do with human emotions such as anger, sadness, or excitement. The ‘mood’ of a verb refers to how the writer presents an idea. The three moods are:

“Indicative mood is the one most often used. In general, it is used for situations when facts and reality, … are the content of a sentence or clause...

“Imperative [mood] forms direct commands...

“Subjunctive mood generally signals that the action or state specified by the verb is the object of a wish, a hope or fear, a request, a conjecture, ...

“[e.g.] Margaret insists that he take the dog for a walk. (request)"
**Principle Act ‘moods’**

**definition** (DEF) – Definition of an act, formerly a “master file” *(subjunctive)*

**intent** (INT) – an intention to plan or perform an act *(imperative)*

  - **request** (RQO) – a request or order for a service from a request “placer” to a request “fulfiller” *(imperative)*

  - **promise** (PRMS) – intent to perform that has the strength of a commitment *(imperative)*

**event** (EVN) – an act that actually happens, includes the documentation (report) of the event *(indicative)*

**Critical Note** –

“Mood” is not a status code.

Each instance of the Act class may have one and only one value for ‘mood.’

Thus, an act in “order” mood that orders an act in definition mood and results in an Act in ‘event’ mood are three different acts, related through the act relationship.
Example of classCode & moodCode attributes

Abstract

Act
- classCode : CS = ??
- moodCode : CS = ??
- id : II = ??
- otherAttributes

Type known Mood abstract

Observation
- classCode : CS = OBS
- moodCode : CS = ??
- id : II = ??
- otherAttributes

ObservationDefinition
- classCode : CS = OBS
- moodCode : CS = DEF
- id : II = 123
- otherAttributes

ObservationRequest
- classCode : CS = OBS
- moodCode : CS = RQO
- id : II = O-02-35
- otherAttributes

ObservationEvent
- classCode : CS = OBS
- moodCode : CS = EVN
- id : II = 7986
- otherAttributes

Defines a specific kind of observation

Orders a defined kind of observation to be performed

Performs the defined observation to fulfill the order

instantiates

fulfills
Specialization by restriction (constraint)

**Act**
class_cd <= ACT
mood <= ActMood
code <= ActCode

**Observation**
class_cd <= OBS
mood <= ActMood
code <= ObservationType

**LabOrder**
class_cd <= OBS
mood <= RQO
code <= LabObservation

**LabOrder (US)**
class_cd <= OBS
mood <= RQO
code <= LOINC

**LabOrder (UK)**
class_cd <= OBS
mood <= RQO
code <= SNOMED Lab

**ObservationOrder**
class_cd <= OBS
mood <= RQO
code <= ObservationType
“Turning the Crank”

- Reference Information Model
- Vocabulary Domain Spec’s
- Data type Specification
- Domain Requirements

Refinement & Constraint

Implement’n Specification for XML

Domain Information Model

Implement

Abstract Message Definition

XML Schema
Version 3 Principles

- Base all standards on HL7-defined models & vocabulary
- Express user requirements as information model(s) of the domain
- Standards are technology-independent abstract definitions
- Provide standard mapping specifications to represent the abstract definitions in one or more implementation technologies
The “essence” of Version 3

• Apply the ‘best practices’ of software development to developing standards – a model-based methodology
• Predicate all designs on two semantic foundations – a reference information model and a complete, carefully-selected set of terminology domains
• Require all Version 3 standards to draw from these two common resources
• Use software-engineering style tools to support the process.
The problem

Storyboard: A clinician, using a local medical office support system, orders a lab test for one of her patients. The test will be performed on a specimen collected at her office. She will send the specimen by courier, and expects to receive a confirmation that the test will be performed, and a result of the test.
What do we need?

- **Act** *(order)* – Order
  - Participation – Author
    - Role - Physician
      - Entity - Dr. Smith in the MD office
    - Participation – Performer
      - Role – Laboratory
        - Entity - The lab that will perform the test
    - Participation – Subject
      - Role – Specimen
    - Participation – Record target
      - Role – Patient in whose record the result goes
  - Act relationship – Definition
    - Act *(definition)* – Ordered Test

... of this order.

... was the author ...

... in her role as a physician ...

Suzy Smith ...
HL7 Graphic representation

• Foundation of HL7 modeling is defined in a UML profile, but we use an alternate graphic representation that better represents our “messages”

• Persistent elements are in “square boxes”
  – All attributes shown
  – All constraints shown on the diagram

• Phrases are shown as “Arrows” linking elements
  – All attributes shown
  – All constraints shown on the diagram
Building messages – persistent elements

An Act constrained:
by classCode to be an observation,
by moodCode to be an order

Common (reusable) types represent roles of patient, specimen, healthcare workers

An Act constrained:
by classCode to be an observation,
by moodCode to be a definition.
Building messages – contextual phrases
Building messages – relational phrase

The **author** of the **order** is an **assigned entity role** (healthcare worker)

The **subject** of the **observation** is a **specimen role**
Automated serializing of graphic design
Automated serializing of graphic design
Automated serializing of graphic design
Our example schema

The subject of the order is a specimen
The essence of Version 3

• A family of specifications

• Built upon a single model of
  – How we construct our messages
  – The domain of discourse
  – The attributes used

• Constructed in a fashion to rapidly develop a comprehensive, fully constrained specification in XML
The HL7 Vocabulary Technical Committee
Committee Missions

Vocabulary TC:
To identify, organize and maintain coded vocabulary terms used in HL7 standards.

Modeling & Methodology TC:
Responsible for creating and maintaining the HL7 development methodology, and maintaining a Reference Model that reflects the shared models that are developed and used by the HL7 Functional Committees.
HL7 Vocabulary Development Strategy

- Reference existing vocabularies
  - SNOMED CT, LOINC, RxNorm, FDA identifiers
- Collaborate with other SDO’s
  - NCPDP, DICOM, ASTM, X12, CEN, ISO, OMG
- Collaborate with NCVHS Patient Medical Record Information (PMRI) standards and Consolidated Health Informatics (CHI) standards
- Add value by creating linkage between HL7 messages and existing vocabularies
- Only add items that do not already exist
- Collaborate with vocabulary developers to add needed content to existing vocabularies
Registration of Code Systems

• Code systems used in HL7 standards need to be registered

• Registration process
  – Form for Proposing a Code System
  – Form is received and reviewed in Vocabulary TC
  – If there are no objections, code system becomes official registered during the next meeting

• OID (object identifier) is assigned to the new code system (more about OIDs later)
Coordination with US Standards Adoption

• National Committee on Vital and Health Statistics
  – Makes recommendations to the Secretary of HHS
  – HIPAA legislation – make recommendations on Patient Medical Record Information (PMRI) standards

• Consolidated Health Informatics (CHI)
  – Consortium of US government departments and agencies

• NLM contract with HL7
  – Adoption of CHI terminologies within HL7 messages
  – Exchange of a complete EHR
Vocabulary Recommendations

- Core Terminologies
  - SNOMED-CT
  - LOINC (lab only)
  - US National Drug Terminologies
    - RxNorm – “Clinical Drugs”
    - NDF-RT: Mechanism of action and physiologic effects
    - FDA: Ingredient name, manufactured dosage form and package type
The HL7 Vocabulary Model
HL7 Vocabulary Model

• The underlying model for the HL7 vocabulary is represented in UML in the HL7 CTS specification.
• Describes the relationship between HL7 coded attributes and vocabulary.
• Based on the [HL7_V3_Meta-Model_Version_1.16](#)
• Primary purpose is to describe the classes and relationships that have a direct bearing on the contents of HL7 coded attributes from the perspective of a meta-model.
HL7 Vocabulary Model

• Divided into:
  – Code Systems
  – Vocabulary Domains
  – Value Sets
  – Coded concepts
  – Concept Designation
  – Concept Property
  – Concept Relationship
HL7 Vocabulary Model - Code System

• A set of unique codes that represent corresponding set of classes in the ‘real world’.

• At various times referred to as an ontology, classification, terminology, or code set.

• Within the HL7 context, concept codes within a code set must not change ‘meaning’.
  – Codes may be added or retired
  – Definitions may be clarified
  – New relationships may be established
  – Codes may not be reused.

• Changing the meaning of concept code(s) results in creation of a new code system.
Code systems may vary in size and complexity from a simple code/value table…

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Male</td>
</tr>
<tr>
<td>F</td>
<td>Female</td>
</tr>
<tr>
<td>U</td>
<td>Undifferentiated</td>
</tr>
</tbody>
</table>

… to a complex reference terminology containing many 100,000’s of concepts, relationships and the like.
Code System Examples

- LOINC
- CPT-4
- NIC
- NOC
- ICD-9-CM
- ICD-10
- SNOMED International
- SNOMED-CT
- ISO 4217 Currency codes
- ISO 3166-2 Country Codes
- IETF Mime Types
- HL7 Version 2 Table 1
- ISO 639 Language Codes
- International Airport Codes
- IANA Character Sets
- HL7 Version 3
- Administrative Gender
- HL7 Version 3 Code System Identifiers
- …
External Code Systems – HL7’s policy is to use existing code systems whenever possible. HL7 will not develop their own code system unless all external possibilities have proven unworkable

- Externally Maintained - HL7 references the contents of the code system but does not maintain or distribute content. (e.g. LOINC)
- Internally Maintained - HL7 maintains an image of the contents for the convenience of its members (e.g. ISO 3166-2 country codes)
Internal Code Systems – code systems that are
developed and maintained within the HL7 organization

- **Structural Codes** - Significant portions of the RIM
  model are represented as concept codes (e.g. ActClass,
  ActCode, EntityClass, etc.)
- **Short Code Lists** - Short tables of codes that are tightly
  coupled with the RIM and have not (as of yet) warranted
  external references (e.g. AdministrativeGender, ActPriority,
  etc.)
- **Version 2 Tables** - Already maintained by HL7 and
  carried in V3 for compatibility.
HL7 Vocabulary Model – Code System

• A **Code System** may define zero or more **Coded Concepts**.

• Every **Coded Concept** must be defined in exactly one **Code System**.

• A **Code System** may represent zero or one **CodeSystem Version** at any given point in time.
HL7 Vocabulary Model – Vocabulary Domain

- A vocabulary domain serves as the link between an HL7 coded attribute and the set(s) of valid concept codes for that attribute.
- A vocabulary domain represents an abstract conceptual space such as "countries of the world", "the gender of a person used for administrative purposes", etc.
HL7 Vocabulary Model – Value Set

• A list of valid concept codes is referred to as a **Value Set** that represent a **Vocabulary Domain**.

• A **Vocabulary Domain** may be represented by zero or more **Value Sets**.

• A **Value Set** may include a list of zero or more **Coded Concepts** drawn from a single **Code System**.

• A **Value Set** may include a list of zero or more **Coded Concepts** drawn from a single **Code System**. A **Value Set** can represent:
  – All of the **Coded Concepts** defined in exactly one **Code System**
  – A specified list of **Coded Concepts** that are defined in exactly one **Code System**
  – The set of **Coded Concepts** represented by another **Value Set**.
A Coded Concept is unique within the Code System that defines it.

Coded Concepts may be characterized by zero or more Concept Properties.

A Coded Concept has the following attributes:
- code - an identifier that uniquely names the class or "concept" within the context of the defining Code System.
- status - represents the current status of the Coded Concept within the Code System.
Code System Identifiers (OIDs)

- OID – ISO Object Identifier
- Sequence of integers representing a Registration Authority tree
  - “…a convenient mechanism for assigning world-unique identifiers to standard-related objects”\(^1\)
  - **Not** a directory tree of entities or objects
  - New entries can be registered in a de-centralized fashion

- [http://www.alvestrand.no/objectid/top.html](http://www.alvestrand.no/objectid/top.html)
- [http://www.iana.org/assignments/enterprise-numbers](http://www.iana.org/assignments/enterprise-numbers)

\(^1\) *Network and Distributed Systems Management*, Sloman
How to get your own OID

Go to the HL7 home page:

http://www.hl7.org

Click on “OID Registry” under the Resources column

Select “Request an HL7 OID”, and follow out the instructions.
HL7 Vocabulary Model

• More details (including detailed UML representations) can be found in the CTS specification at:

http://informatics.mayo.edu/LexGrid/downloads/CTS/specification/ctsSpec/cts.htm#CTSCommErrMsgRTAPI
Coded Data Types for Version 3

For further details see:
Data Types Abstract Specification
and
V3 XML Data Types Implementation Technology Specification (ITS)
Coded Data Types (Ballot Package Mar ‘04)
XML Schema for Concept Descriptor

```xml
<xsd:complexType name="CD">
  <xsd:complexContent>
    <xsd:extension base="ANY">
      <xsd:sequence>
        <xsd:element name="originalText" type="ED" ... />
        <xsd:element name="qualifier" type="CR" ... />
        <xsd:element name="translation" type="CD" ... />
      </xsd:sequence>
      <xsd:attribute name="code" type="cs" ... />
      <xsd:attribute name="codeSystem" type="uid" ... />
      <xsd:attribute name="codeSystemName" type="st" ... />
      <xsd:attribute name="codeSystemVersion" type="st" ... />
      <xsd:attribute name="displayName" type="st" ... />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```
type CodedSimpleValue alias CS specializes CV {
  ST code;
  literal ST;
};

<mood_cd code="INT"/>

Code **must** come from the specified HL7 domain
CV – Coded Value

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>ST</td>
<td>The plain code symbol defined by the code system. For example, &quot;784.0&quot; is the code symbol of the ICD-9 code &quot;784.0&quot; for headache.</td>
</tr>
<tr>
<td>code System</td>
<td>UID</td>
<td>Specifies the code system that defines the code.</td>
</tr>
<tr>
<td>codeSystem Name</td>
<td>ST</td>
<td>The common name of the coding system.</td>
</tr>
<tr>
<td>codeSystem Version</td>
<td>ST</td>
<td>If applicable, a version descriptor defined specifically for the given code system.</td>
</tr>
<tr>
<td>displayName</td>
<td>ST</td>
<td>A name or title for the code, under which the sending system shows the code value to its users.</td>
</tr>
<tr>
<td>originalText</td>
<td>ED</td>
<td>The text or phrase used as the basis for the coding.</td>
</tr>
</tbody>
</table>

```csharp
type CodedValue alias CV specializes CE {
    ST code;
    UID codeSystem;
    ST codeSystemName;
    ST codeSystemVersion;
    ST displayName;
    ED originalText;
};
```
Example -- Coded Value (HL7 code system)

Patient Gender

**XML**

```
<administrative_gender_cd
code="M"
displayName="Male"
codeSystem="2.16.840.1.113883.5.1"
codeSystemName="Gender:USA:HL7"
codeSystemVersion="3.0"/>

<gender_cd code="M"
codeSystem="2.16.840.1.113883.5.1"/>
```
CE – Coded with Equivalents

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>ST</td>
<td>The plain code symbol defined by the code system. For example, &quot;784.0&quot; is the code symbol of the ICD-9 code &quot;784.0&quot; for headache.</td>
</tr>
<tr>
<td>codeSystem</td>
<td>UID</td>
<td>Specifies the code system that defines the code.</td>
</tr>
<tr>
<td>codeSystem Name</td>
<td>ST</td>
<td>The common name of the coding system.</td>
</tr>
<tr>
<td>codeSystem Version</td>
<td>ST</td>
<td>If applicable, a version descriptor defined specifically for the given code system.</td>
</tr>
<tr>
<td>displayName</td>
<td>ST</td>
<td>A name or title for the code, under which the sending system shows the code value to its users.</td>
</tr>
<tr>
<td>originalText</td>
<td>ED</td>
<td>The text or phrase used as the basis for the coding.</td>
</tr>
<tr>
<td>translation</td>
<td>SET&lt;CD&gt;</td>
<td>A set of other concept descriptors that translate this concept descriptor into other code systems.</td>
</tr>
</tbody>
</table>

Type CodedWithEquivalents alias CE specializes CD {
    ST code;
    UID codeSystem;
    ST codeSystemName;
    ST codeSystemVersion;
    ST displayName;
    ED originalText;
    SET<CV> translation;
};
Patient Blood Type

**XML**

```xml
<bloodTypeCode
code="C0302037"
displayName="Blood Group Antigen A, NOS"
codeSystem="2.16.840.1.113883.6.56"
codeSystemName="UMLS"
codeSystemVersion="2003AC"
codingRationale="HL7">
  <translation code="14711003" codeSystem="2.16.840.1.113883.6.5" codeSystemName="SCT" codeSystemVersion="January 2004 Release"
codingRationale="SRC"/>
</bloodTypeCode>
```

This example assumes that the UMLS Metathesaurus was chosen as the preferred scheme, but the sending system was using SNOMED.
## CD – Concept Descriptor

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>codeSystem</td>
<td>UID</td>
<td>Specifies the code system that defines the code.</td>
</tr>
<tr>
<td>codeSystemName</td>
<td>ST</td>
<td>The common name of the coding system.</td>
</tr>
<tr>
<td>codeSystemVersion</td>
<td>ST</td>
<td>If applicable, a version descriptor defined specifically for the given code system.</td>
</tr>
<tr>
<td>displayName</td>
<td>ST</td>
<td>A name or title for the code, under which the sending system shows the code value to its users.</td>
</tr>
<tr>
<td>originalText</td>
<td>ED</td>
<td>The text or phrase used as the basis for the coding.</td>
</tr>
<tr>
<td>translation</td>
<td>SET&lt;CD&gt;</td>
<td>A set of other concept descriptors that translate this concept descriptor into other code systems.</td>
</tr>
<tr>
<td>qualifier</td>
<td>LIST&lt;CR&gt;</td>
<td>Specifies additional codes that increase the specificity of the the primary code.</td>
</tr>
<tr>
<td>codingRationale</td>
<td>CS</td>
<td>Identifies how to interpret the instance of the code.</td>
</tr>
</tbody>
</table>

```plaintext
type ConceptDescriptor alias CD specializes ANY {
  ST code;
  ST displayName;
  UID codeSystem;
  ST codeSystemName;
  ST codeSystemVersion;
  ED originalText;
  LIST<CR> qualifier;
  SET<CD> translation;
  CS codingRationale; ... };
```
**Definition:** A concept qualifier code with optionally named role. Both qualifier role and value codes must be defined by the coding system of the CD containing the concept qualifier. For example, if SNOMED RT defines a concept "leg", a role relation "has-laterality", and another concept "left", the concept role relation allows to add the qualifier "has-laterality: left" to a primary code "leg" to construct the meaning "left leg".

protected type ConceptRole alias CR specializes ANY {
    CV name;
    BN inverted; CD value;
}

Use for roles like: has laterality, has body location, etc.
Vocabulary Binding

Marrying the Information Models to the Terminologies
Vocabulary Binding

• The point at which vocabulary models and information models meet is referred to as Vocabulary Binding.

• In HL7, the binding of attributes in a RIM-derived, reusable static model to specific vocabularies is the point where these models interact.

• Vocabulary and information model interaction is not a black and white concept.
  – In one case you may need to create a Value Set to constrain the allowable values for a concept, and bind the model to that Value Set.
  – In another, you can use the model constraint rules to enumerate a ‘virtual’ value set.
Vocabulary Binding - Recommendations

- Attributes and Datatype Properties are bound to vocabulary domains in the RIM at the time the RIM is defined whenever the data type of the attribute or property is derived from CD
Vocabulary Binding - Recommendations

• Domains can be associated with Value Sets on a realm-specific basis.

• Some domains are defined as "non-realm-specializable" which means that only one value-set can be bound to that domain and realms are not permitted to define separate bindings.

• All "non-realm-specializable" domains *must* have a value-set binding.
Vocabulary Binding - Recommendations

• In static models (D-MIMs, R-MIMs, HMDs, Message Types, Templates, Static Profiles, etc.) attributes and properties can be either bound to vocabulary domains or value-sets
  – Vocabulary Domains are abstract concepts which can be bound to distinct value-sets by realm (such as Canada, Japan or US)
  – Value-sets are collections of specific codes from one or more code systems where the codes within a value-set each have distinct meaning
Vocabulary Binding - Recommendations

• When constraining models (see static model discussion above) a designer has five options:
  – a) Leave the child element vocabulary domain the same as with the parent model vocabulary domain
  – b) Constrain the 'child' vocabulary domain to a 'child' domain which represents a narrower concept than the parent vocabulary domain
  – c) Constrain the vocabulary domain to a specific value-set
d) Leave the 'child' value-set the same as the parent value-set
e) Constrain the 'child' value-set to a narrower value-set than parent value-set
Vocabulary Binding - Recommendations

• In some circumstances, domains and/or Value Sets may be asserted as 'free-hand constraints' rather than a direct binding of a single domain to the attribute or property.

• If so, each domain and/or Value Set that is asserted by the constraint must follow the rules defined by 4.
• Constraining to Value Sets should happen as late in the constraint process as possible to encourage the general application/use of the artifacts so constrained.
Vocabulary Binding - Recommendations

• There are constraints on when a vocabulary domain can be constrained to a Value Set
  – a) A non-realm-specializable domain can always be constrained to a Value Set (on the grounds that a domain always has a 1..1 relationship to a Value Set)
  – b) A non-universal domain can only be constrained to a Value Set if the model is itself bound to a single realm and that realm has bound the parent domain to a Value Set.
  – c) The Value Set bound to a child whose parent is a domain must either be the same as the Value Set tied to the parent based on realm, or a subset thereof.
• When validating an instance, the Value Set to be bound against should be determined as follows:
  
  – a) The allowed Value Set is the intersection of the Value Sets of the balloted static model for the interaction transmitted as well as any profiles and templates which apply to the scope of the element being validated
  
  – b) Where the balloted static model or templates/profiles bind to a domain, the value-set is determined by the realm that is effective at that point in the instance
Common Terminology Services
Common Terminology Services (CTS)

• An HL7 ANSI approved standard Terminology Service
• Purpose is to specify a common Application Programming Interface (API) to access terminological content.
• Developed as an alternative to a common data structure.
• Identifies the common functional characteristics that an external terminology must be able to provide.
• Client software doesn’t have to know about specific terminology data structures and/or how to access them.
• Server software can plug and play with many clients.
Find codes having “*myelitis”

Select * from VOC_concept_designation
WHERE text like ‘%myelitis’
Common Terminology Services API

- Allows Client Software to be Developed Independently from Service Server Software
- Allows Terminology Plug-and-Play
- Allows Client Plug-and-Play
- Defines a “Functional Contract”
Additional CTS API’s

• CTS Message Browsing API
  – Used by HL7 Modelers

• CTS Vocabulary Browsing API
  – Used by HL7 Terminology Authors and Value Set Building

• CTS Mapping API
  – Used to translate concept codes from one system to another

Details can be found on HL7 Ballot spec
Common Terminology Services

• **Interface specification**
  – Different message processing applications, same functions
  – Different terminology structures, philosophy – same behavior

• **Language Bindings**
  – (Currently) specified in OMG IDL
  – Java interface binding
  – Java bean binding
  – WSDL/SOAP binding

• **Version 1.0 Finalized Spring 2004**
Common Terminology Services

Resources:

• **Specification:**

• **Implementations:**
  – [http://informatics.mayo.edu/](http://informatics.mayo.edu/)
    - Tools / CTS Demos - open source SOAP, Java and JSP implementations of CTS

• **CTS Implementer's Mail Server**
  – cts-impl@kestral.com.au
  – To get someone added to the list, send an email to either grahame@kestral.com.au or jamies@kestral.com.au

• **CTS Implementer's WIKI**
Common Terminology Services – Future Work

• Mayo Clinic reference implementation proposed for use by:
  – The NHS from the UK
  – Eclipse OHF,
    • Users are looking to OHF to provide a CTS implementation under EPL.

• CTS II
  – New project within the Vocab TC
  – Additional CTS functionality
    • Import
    • Edit/Update
    • OWL
    • Extended H7 Functionality – value set constraints, etc.
Thank You!