BRIDG Model

http://www.bridgmodel.org

BRIDG Modeling and Naming Conventions

Prepared by BRIDG Semantic Coordination Committee (SCC)

Becky Angeles (ScenPro/NCI [C]/HL7 RCRIM)
Julie Evans (CDISC/HL7 RCRIM)
Smita Hastak (ScenPro/NCI [C])
Lloyd McKenzie (Gordon Point Informatics/NCI [C]/HL7)
Charlie Mead (NCI [C]/HL7)
Wendy Ver Hoef (ScenPro/NCI [C])

September 6, 2013

BRIDG Biomedical Research Integrated Domain Group

Table of Contents

Revision History

Version Number	Date	Summary of Changes
1.0	June 13, 2011	First official version with SCC comments incorporated
1.1	Aug. 5, 2011	Incorporated more SCC comments
1.2	Aug. 11, 2011	Incorporated more SCC comments and title page
1.3	Sept. 6, 2011	Updated title page and page header, added second paragraph re ISO 21090
1.4	Nov. 15, 2011	Added Modeling Convention and Pattern text for Key Distinctions Between Purpose, Objective and Reason
1.5	Feb. 29, 2012	Added two new constraints to the Constraints section. Updated Model Element Description section – provided template for attribute definitions for some common attributes types; Moved few sections into the User Guide; Updated the reference of ISO data types to HL7 Abstract Data Type Release 2 in support of BRIDG r 3.1
1.6	Aug. 7, 2012	Added section dealing with mandatory attributes
1.7	Aug. 30, 2012	Updated title page
1.8	Sep 6, 2013	Made document 508 Compliant

Overview

This document describes the modeling conventions used by the BRIDG SCC when representing semantics in the BRIDG Model. It identifies patterns used in the model, naming conventions for various model elements, and best practices for creating good definitions.

Modeling Conventions and Patterns

Modeling Business Rules Using Constraints

To provide a visual clue that there is a business rule that applies to the semantics represented by a class, the BRIDG SCC models business rules about relationships, attribute values and other kinds of logical constraints as Enterprise Architect (EA) constraints at the class level (attribute level constraints are not visually represented in EA):

We have identified the following types of constraints to date:

- **Exclusive Or** – used to specify that only one of several similar associations can be used for any given instance of the class
- **Not Applicable** – used to indicate that an attribute or association is not used in a certain context such as on a subclass that inherits an association that applies to the superclass but not this subclass
- **Qualifier** – used to restrict a value on an attribute, to require or disallow a certain association in a specific situation, and generally capture other kinds of business rules (e.g. on PerformedEligibilityCriterion the "is participated In By Qualifier" constraint reads: "Only the association from StudySubject is valid for PerformedEligibilityCriterion and its subclasses" meaning that the association from ExperimentalUnit is not valid.)
- **Unique** Qualifier – used to indicate that the value of a single attribute should be unique within a given context
- **Attribute Set Unique Qualifier** – used to indicate that the values of a set of 2 or more attributes should be unique within a given context
- <association label> actualIndicator Qualifier - used to indicate that an entity on the other end of the association must be a "kind of" or "instance of" entity
- Attribute Set actualIndicator Qualifier – used to indicate when a set of attributes is valid for only an "instance of" entity.
- **Declaration** - used to highlight the fact that for any subclasses, inherited properties or associations are valid unless explicitly constrained or excluded at the subclass level

Describe the business rule as clearly as possible and include complete references to attributes in classes other than where the constraint is defined.

See Naming Conventions below for constraint names.

Identifier Types

When a given model concept has the possibility of being assigned more than one identifier from different sources and/or purposes, there is a need to distinguish one identifier from another by type. For example, a person may be assigned a medical record number (MRN) as well as a social security number (SSN), driver's license number, and insurance number. The HL7 Abstract data type for Instance Identifier (II) is only meant to be a globally unique identifier and intentionally does not identify type – even the data type attribute II.identifierName is not to be used to computably determine the type, source or any kind of meaning for the identifier. Thus the pattern for identifiers that require identification of a type is to move the identifier attribute into its own class and add a typeCode attribute to capture the type of that identifier with a one-to-many association between the original class and the identifier class. Additionally there may be an association between the identifier class and the assigning entity if needed (e.g. organization, study registry, or system of record).

Automated Coding and Modified Text

Some concepts in the BRIDG Model are represented by a code that is determined by an automated coding application. Automated coders translate the original text into a code if the original text is something it recognizes and can parse. However, if there is a typo or unknown wording, etc., the auto-coders are unable to process the original text. Human intervention is required to interpret the text and provide corrected or modified text to allow the auto-coder to do its job.

However, the HL7/ISO data type called Concept Descriptor (CD) does not include an attribute for such modified text. Thus in cases where the BRIDG Model captures a concept that may be auto-coded, the class also includes a modified text attribute to capture the corrected text. Examples of this in the BRIDG Model are Product.codeModifiedText and PerformedObservationResult.valueCodeModifiedText.

Mandatory Attributes

Because the BRIDG model may be used for any number of purposes, the number of mandatory attributes is minimized as much as possible to prevent undue burden on users of the BRIDG model. Users are encouraged to constrain the subset of classes they use to meet their business needs. The only cases identified so far, fall in one or more of the following categories:

- The class doesn't make sense without that attribute having a value, for example:
  - BiologicEntityIdentifier.identifier – this attribute is essential to the purpose of the class;
  - StudyCondition.code – there is no meaning to this class if this attribute has no value;
- The semantic is ambiguous without that value specified, for example:
  - Material.actualIndicator, Organization.actualIndicator or any class with an actualIndicator – it's essential to know whether we're talking about an actual thing or a kind of thing;

- The attribute value is necessary to be able to distinguish one instance from another, for example:
  - SystemOfRecord.name – this is the only attribute and no associations can be used to identify it otherwise;
  - DefinedExpressionVariableRelationship.localVariableName - this is the only attribute and associations can't be used to identify it otherwise.

Essentially, if ever there's a conceivable use case that needs the class, but not the attribute, the attribute must not be mandatory.

Note that when an association can be used to (help) uniquely identify something or if two or more alternates might provide enough meaning or identity instead of a specific single attribute, then BRIDG uses a class-level constraint to specify "mandatoriness", for example:

- QualifiedPerson has a *Mandatory Attribute Qualifier* constraint: "At least one of the following attributes must be mandatory for a QualifiedPerson: typeCode, certificateLicenseText."
- Arm has a *Mandatory Attribute Qualifier* constraint: "At least one of the following attributes must be mandatory for Arm: name, typeCode."

Derived Data

In many models that are harmonized with BRIDG, there are model elements that are, from a BRIDG perspective, considered derived. The BRIDG SCC generally makes the case for these semantics to not be added to the BRIDG model, but if the domain experts have a strong tie to specific terms that describe the concept or if the source data from which the derived data is generated is not collected or available (i.e. the data is only collected in derived form and the source data is not also collected), the BRIDG SCC will add derived attributes and explain in the Notes section of the attribute descriptions how the data is derived. But when the project team is satisfied with documenting in the mapping spreadsheet how the data is derived from a BRIDG perspective, then attributes are not added to the BRIDG Model.

Demographic Data

There are several ways that new demographic data can be represented in the BRIDG Model. The two main ways are as observations (DefinedObservation, PerformedObservation) or as attributes of BiologicEntity, Person, Subject or StudySubject. Choosing which way to go is ultimately at least a somewhat subjective decision, however the following considerations can help determine which option is a better fit.

- **Static vs Dynamic** – Does the demographic characteristic change over time? If so, it likely makes sense to model it as an observation. If not, then it might be better represented as an attribute of BiologicEntity, Person, Subject or StudySubject (henceforth referred to collective as BiologicEntity-related classes). For example, Person.raceCode is generally considered static, while Body Mass Index (BMI) is dynamic. This is not a strict distinction as BRIDG has historically made maritalStatusCode and primaryOccupationCode attributes of Person due to how commonly used they are, but

for new attributes this consideration is a good indicator of which modeling approach to use.

- **Common vs Rare** – Is this demographic characteristic commonly collected by most studies or is it very study-specific? If it's only rarely collected, then it's best modeled as an observation. Only if it's extremely ubiquitous should it be modeled as an attribute of a BiologicEntity-related class. For example, sex or gender (BiologicEntity.administrativeGenderCode) is almost always collected, but eye color is rarely collected.
- **Administrative vs Research** – What is the purpose of the data? If it's administrative information, such as how study expenses will be covered for a subject, then it makes sense to model it as an attribute of a BiologicEntity-related class. If it's study data that will be analyzed as part of the research objectives and measures, then it's best modeled as an observation. For example, StudySubject.paymentMethodCode is often considered administrative in nature, whereas BMI would be collected for research purposes.
- **Reported vs Observable** – Is the data reported by the subject and not observable by others, then it might be an attribute of a BiologicEntity-related class, whereas if it's an observable characterstic, it's probably best modeled as an observation. For example, Person.primaryOccupationCode is not an observable attribute, but could only be reported by the subject, but eye color is observable.

Model Layout and Display

Sub-Domain and Comprehensive Class Diagrams

The BRIDG Model breaks up the set of classes into sub-domain packages. Each package has at least one class diagram showing the classes in that sub-domain and any other relevant classes. These diagrams present a subset of the BRIDG semantics in smaller groups which are more digestible for users and are focused on specific topics. While it is sometimes helpful to break up classes into smaller diagrams like this, it is also extremely helpful to have a comprehensive diagram that has every class, attribute and association displayed to ensure that overall model integrity is maintained. This prevents or at least helps identify cases when duplicate or unnecessary associations are created and it ensures that related concepts have the necessary links.

Displaying Constraints and Inherited Attributes

For model integrity purposes, the BRIDG model always displays constraints and inherited attributes. This prevents accidentally adding duplicate attributes and ensures that the full context of any given class is readily available. Additionally, displaying constraints gives a visual clue that there is something constraining the data that will be captured by the class.

Color Coding of BRIDG Classes

BRIDG classes are colored based on their sub-domain groupings. Classes in the Adverse Event sub-domain are green, Common classes are blue, Protocol Representation classes are purple, Regulatory classes are orange, and Study Conduct classes are pink.

Diagram Layout

In general, BRIDG has located classes in the class diagram with other classes to which they are related. More specifically, the following conventions are generally followed:

- Entities (e.g. people, places, things) are grouped on the left side of the diagram and activities are grouped on the right side with roles and classes that associate entities and activities between them.
- When possible, subclasses are placed below their super-class and classes with a 0..* association to another class are placed to the right of or below the other class.
- Whenever possible, avoid crossing lines and close lines, maintaining some whitespace in the diagram for readability.
- Diagrams with spaghetti associations are difficult to read and even more difficult to use or validate, so the BRIDG SCC recommends "Manhattanizing" the diagram by making all the associations run horizontally or vertically. This can be done using the Control-Shift-A feature in Enterprise Architect which auto-routes the association. However it is done, the goal is to make the diagram readable.
- The placement of association labels should be considered to avoid placing them on the intersection of 2 or more lines.

Notes in Diagrams

Enterprise Architect supports two different kinds of notes that may be useful when building BRIDG-based models:

- A diagram note and its properties should be displayed in the top left corner of the diagram.
- Regular notes can be added to help identify questions to review with team members, but generally should not remain on a finished model.

Warning: Notes added to a diagram and associated to a class or association do not get replicated to other diagrams in which that class or association appears.

Naming Conventions

The naming conventions described below are general guidelines and should be followed unless sufficient justification exists. That are not meant to be slavishly followed if the semantics would be poorly represented by following them, but exceptions are and should be rare.

Class Names

- Are a singular noun composed of one or more words
- Use precise words and avoid vague or ambiguous words
- Begin with a capital letter and use camelCase for the rest of the name if there is more than one word in the name
- Do not contain spaces, conjunctions or prepositions
- Avoid use of abbreviations, acronyms and jargon
- Are unique throughout the model (even across packages)
- Avoid repeating the super-class name entirely in the subclass name

Attribute Names

- Are a singular noun composed of one or more words
- Use precise words and avoid vague or ambiguous words
- Begin with a lowercase letter and use camelCase if there is more than one word in the name
- Do not contain spaces, conjunctions or prepositions
- Avoid use of abbreviations, acronyms and jargon
- Are unique throughout the class
- Avoid including the class name in the attribute name, unless necessary for domain recognition
- Include the following suffixes as appropriate:
  - Indicator – used for Boolean data types
  - Date – used for single dates
  - DateRange – used for a period of time
  - Code – used when data should be coded, even if example data is not coded yet (can use the CD.originalText attribute for uncoded data)
  - Name
  - Text – used when the domain term for the concept implies a non-text data type but can contain textual characters, such as lot number, which can contain letters, not just numbers and is therefore named lotNumberText

Association Names

- Are a single verb or verb phrase describing the semantics of the relationship between source and target
- Are complimented by an association level constraint describing the semantics of the relationship between target and source (the inverse direction)
- Default to “specializes” for generalization relationships
- Default to “is collected in” for aggregation relationships

- Default to "is part of" for composite relationships

Constraint Names

- Are named after the model element (attribute or association) which is being constrained
- Include a descriptor after the model element, typically one of the following constraints: Exclusive Or; Not Applicable; Qualifier; Unique Qualifier; Attribute Set Unique Qualifier; Declaration; ActualIndicator Qualifier; Attribute Set actualIndicator Qualifier *(See page 3 of this document for description of each of these constraints)*
- Some example patterns for constraint text for actualIndicator constraints:
  - For <association label> actualIndicator Qualifier – Only <ClassName1 or ClassName2 . . > with actualIndicator = "[true|false]" is valid
  - For Attribute Set actualIndicator Qualifier: <attribute names> are valid only when actualIndicator = "true".

Model Element Descriptions

Include the following sections for class and attribute descriptions:

- Definition – this should be a brief statement of the meaning of the class or attribute, using singular terms, and not using the words embedded in the name of the element. Explanations should be moved to the Notes section
- Examples – unless the attribute is capturing dates or numbers, including valid domain examples is very helpful in clarifying understanding if the definition is less than ideal
- Other Names – include aliases used by other organizations, domains, etc.
- Notes – include any explanations, rationale for the concept, etc.

Use templates for definitions for attributes when applicable (refer to Appendix A below for additional definition guidelines)

- xxxIndicator: Specifies whether the ….
- xxxCode: A coded value specifying the…
- xxxDate: The date (and time) on which …
- xxxDateRange: The date and time span for when …
- xxxName: A non-unique textual identifier …
- xxxText: A character string …
- identifier: A unique symbol that establishes identity of …
- typeCode: A coded value specifying the kind of …
- statusCode: A coded value specifying the phase in the lifecycle of …

Associations have definitions that fit the following pattern:

- "Each SourceClass [always | might] xxxxxxx [one | one or more] TargetClass. Each TargetClass [always | might] xxxxxxx [one | one or more] SourceClass."

- Example: “Each OrganizationalContact might be a function performed by one Person. Each Person might function as one or more OrganizationalContact.” Note that definitions for associations read the relationship going both directions.
- Make sure the tense of the definition is appropriate if there is a planned/scheduled/performed nature to the association.

When defining classes and attributes, **good definitions**…

- State the essential meaning of the concept
- State what the concept is, not only what it is not
- State as a descriptive phrase or sentence(s)
- Are appropriate for the type of term being defined
- State in the singular form
- Use the same terminology and consistent logical structure for related definitions
- Contain only commonly understood abbreviations and only when necessary
- Are able to stand alone
- Are expressed without embedding definitions of other data or underlying concepts
- Are precise and unambiguous
- Are concise
- Are expressed without embedding rationale, functional usage, domain information or procedural information (this supplemental information can go in the Notes section)
- Avoid circular reasoning

See Appendix A below for a more full discussion of these principles.

Appendix A: Creating Well-Formed Definitions

State the essential meaning of the concept

All primary characteristics of the concept represented should appear in the definition at the relevant level of specificity for the context. The inclusion of non-essential characteristics should be avoided. The level of detail necessary is dependent upon the needs of the system user and environment.

EXAMPLE 1 - "Consignment Loading Sequence Number" (Intended context: any form of transportation)

1) good definition: A number indicating the sequence in which consignments are loaded in a means of transport or piece of transport equipment.
2) poor definition: A number indicating the sequence in which consignments are loaded in a truck.

REASON - In the intended context, consignments can be transported by various transportation modes, e.g., trucks, vessels or freight trains. Consignments are not limited to trucks for transport.

EXAMPLE 2 - "Invoice Amount"

1) good definition: Total sum charged on an invoice.
2) poor definition: The total sum of all chargeable items mentioned on an invoice, taking into account deductions on one hand, such as allowances and discounts, and additions on the other hand, such as charges for insurance, transport, handling, etc.

REASON - The poor definition includes extraneous material.

State what the concept is, not only what it is not

When constructing definitions, the concept cannot be defined exclusively by stating what the concept is not.

EXAMPLE - "Freight Cost Amount"

1) good definition: Cost amount incurred by a shipper in moving goods from one place to another.
2) poor definition: Costs which are not related to packing, documentation, loading, unloading, and insurance.

REASON - The poor definition does not specify what is included in the meaning of the data.

State as a descriptive phrase or sentence(s)

A phrase is necessary to form a precise definition that includes the essential characteristics of the concept. Simply stating one or more synonym(s) is insufficient. Simply restating the words of the name in a different order is insufficient. If more than a descriptive phrase is needed, use complete, grammatically correct sentences.

EXAMPLE - "Agent Name"

1) good definition: Name of party authorized to act on behalf of another party.
2) poor definition: Representative.

REASON - "Representative" is a near-synonym of the data element name, which is not adequate for a definition.

Are appropriate for the type of term being defined

Different types of terms play a different role and this should be reflected in the definitions.

EXAMPLE - "Job Grade Maximum Salary Amount": The maximum salary permitted for the associated job grade.

Note: Makes no reference to a specific value domain.

EXAMPLE - "Monetary amount": An amount that may be expressed in a unit of currency.

Note: Refers to a "dimensionality" of currency, but not to a specific currency.

EXAMPLE - "European Job Grade Maximum Salary Amount": The maximum salary permitted for the associated job grade expressed in Euros.

EXAMPLE - "U.S. Job Grade Maximum Salary Amount": The maximum salary permitted for the associated job grade expressed in US dollars.

State in the singular form

The concept expressed by the definition shall be expressed in the singular.

EXAMPLE - "Article Number"

1) good definition: A reference number that identifies an article.
2) poor definition: Reference number identifying articles.

REASON - The poor definition uses the plural word "articles," which is ambiguous, since it could imply that an "article number" refers to more than one article.

Use the same terminology and consistent logical structure for related definitions

A common terminology and syntax should be used for similar or associated definitions.

EXAMPLE - The following example illustrates this idea. Both definitions pertain to related concepts and therefore have the same logical structure and similar terminology.

1) "Goods Dispatch Date" - Date on which goods were dispatched by a given party.
2) "Goods Receipt Date" - Date on which goods were received by a given party.

REASON - Using the same terminology and syntax facilitates understanding. Otherwise, users wonder whether some difference is implied by use of synonymous terms and variable syntax.

Contain only commonly understood abbreviations

Understanding the meaning of an abbreviation, including acronyms and initialisms, is usually confined to a certain environment. In other environments the same abbreviation can cause misinterpretation or confusion. Therefore, to avoid ambiguity, full words, not abbreviations, shall be used in the definition. Exceptions to this requirement may be made if an abbreviation is commonly understood such as “i.e.” and “e.g.” or if an abbreviation is more readily understood than the full form of a complex term and has been adopted as a term in its own right such as “radar” standing for “radio detecting and ranging.” All acronyms must be expanded on the first occurrence.

EXAMPLE - “Tide Height”

1) good definition: The vertical distance from mean sea level (MSL) to a specific tide level.
2) poor definition: The vertical distance from MSL to a specific tide level.

Are able to stand alone

The meaning of the concept should be apparent from the definition. Additional explanations or references should not be necessary for understanding the meaning of the definition.

EXAMPLE - “School Location City Name”

1) good definition: Name of the city where a school is situated.
2) poor definition: See “school site”.

REASON - The poor definition does not stand alone, it requires the aid of a second definition (school site) to understand the meaning of the first.

Are expressed without embedding definitions of other data or underlying concepts

The definition of a second data element or related concept should not appear in the definition proper of the primary data element. Definitions of terms should be provided in an associated glossary. If the second definition is necessary, it may be attached by a note at the end of the primary definition's main text or as a separate entry in the dictionary. Related definitions can be accessed through relational attributes (e.g., cross-reference).

EXAMPLE - “Sample Type Code”

1) good definition: A code identifying the kind of sample.
2) poor definition: A code identifying the kind of sample collected. A sample is a small specimen taken for testing. It can be either an actual sample for testing, or a quality control surrogate sample. A quality control sample is a surrogate sample taken to verify results of actual samples.

REASON - The poor definition contains two extraneous definitions embedded in it. They are definitions of “sample” and of “quality control sample.”

Are precise and unambiguous

The exact meaning and interpretation of the defined concept should be apparent from the definition. A definition should be clear enough to allow only one possible interpretation.

EXAMPLE - "Shipment Receipt Date"

1) good definition: Date on which a shipment is received by the receiving party.
2) poor definition: Date on which a specific shipment is delivered.

REASON - The poor definition does not specify what determines a "delivery." "Delivery" could be understood as either the act of unloading a product at the intended destination or the point at which the intended customer actually obtains the product. It is possible that the intended customer never receives the product that has been unloaded at his site or the customer may receive the product days after it was unloaded at the site.

Are concise

The definition should be brief and comprehensive. Extraneous qualifying phrases such as "for the purpose of this metadata registry," "terms to be described," shall be avoided.

EXAMPLE - "Character Set Name"

1) good definition: The name given to the set of phonetic or ideographic symbols in which data is encoded.
2) poor definition: The name given to the set of phonetic or ideographic symbols in which data is encoded, for the purpose of this metadata registry, or, as used elsewhere, the capability of systems hardware and software to process data encoded in one or more scripts.

REASON - In the poor definition, all the phrases after "...data is encoded" are extraneous qualifying phrases.

Are expressed without embedding rationale, functional usage, domain information, or procedural information

Although they are often necessary, such statements do not belong in the definition proper because they contain information extraneous to the definition. If deemed useful, such expressions may be placed in other metadata attributes (see ISO/IEC 11179-3). It is, however, permissible to add examples after the definition.

1) The rationale for a given definition should not be included as part of the definition (e.g. if a data element uses miles instead of kilometers, the reason should not be indicated in the definition).
2) Functional usage such as: "this data element should not be used for ..." should not be included in the definition proper.
3) Remarks about procedural aspects. For example, "This data element is used in conjunction with data element 'xxx'", should not appear in the definition; instead use "Related data reference" and "Type of relationship" as specified in ISO/IEC 11179-3.

EXAMPLE - “Data Field Label”

1) good definition: Identification of a field in an index, thesaurus, query, database, etc.
2) poor definition: Identification of a field in an index, thesaurus, query, database, etc., which is provided for units of information such as abstracts, columns within tables.

REASON - The poor definition contains remarks about functional usage. This information starting with “which is provided for...” must be excluded from the definition and placed in another attribute, if it is necessary information.

Avoid circular reasoning

Two definitions shall not be defined in terms of each other. A definition should not use another concept's definition as its definition. This results in a situation where a concept is defined with the aid of another concept that is, in turn, defined with the aid of the given concept.

EXAMPLE - Two data elements with poor definitions:

1) Employee ID Number - Number assigned to an employee.
2) Employee - Person corresponding to the employee ID number.

REASON - Each definition refers to the other for its meaning. The meaning is not given in either definition.