BRIDG HCT 1.0 Design Decisions

BRIDG HCT Modeling Team:

Charles Martinez
Bob Milius
Jane Pollack
Robinette Renner
Kirt Schaper
Sandra Sorenson
Shyam Somasundaram
Harry Vassilev

Please submit all questions to the corresponding author Jane Pollack (jpollack@nmdp.org)

Document Revision History:

Revision	**Date**	**Last Edited By**
1.0 (Initial)	2015/02/10	Jane Pollack

Who should read this:

We intend this document to be read by database administrators, data architects, and data modelers who may be interested in incorporating some of this content into their own organizations. We wish to use this document as a mechanism to facilitate feedback to the CIBMTR modeling group so that we can better understand the needs and constraints of an IT department that supports the activity of a transplant center. While this is the intended audience of the document, it is written at a level that we hope will facilitate dialog between these IT staff and the organization's stakeholders to create an understanding of what the BRIDG HCT 1.0 model does and does not provide. This document is lengthy, but this is because it is using many diagrams to describe the contents. We hope that this document can be consumed within an hour.

How we structured the document:

We assume the reader has at least a cursory knowledge of BRIDG and the BRIDG model (www.bridgmodel.org). The user's guide provided is a good source of background information. At the time of this writing, the current release is BRIDG 3.2, but work is ongoing to release BRIDG 4.0.

We have broken this document into a roughly chronological format. For example, we describe the different levels of modeling effort and described them in the order at which we engaged in those activities. At each level of this chronology we highlight decision points. We do our best to describe our reasoning for the decision made, and our understanding of the consequences of those decisions. Many of the later decisions are dependent upon the choices that we made at previous levels. We wish for feedback from the community to determine if we should revisit any of those decisions, and its implications to the larger model. In other words we do not intend that the BRIDG HCT 1.0 model is the final model.

We intend to collect feedback from interested parties using het following mechanisms:

1. Tandem 2015 CIBMTR IT Forum on February 12, 2015. This is a one-day event that gathers persons interested in the collection for HCT Outcomes research. We will be gathering feedback at the meeting itself through an online-survey and attendee's comments.
2. Three dialup meetings scheduled for soliciting feedback from those persons who could not attend Tandem. Anyone wishing to attend these meetings should contact the corresponding author to have the invitation forwarded to them.
   a. Thursday, 2015/03/12 @ 11 AM CST
      Review of Design Decision Document and any github issues. Special attention will be made to Unique Identifier Structure.
   b. Thursday, 2015/03/26 @ 11 AM CST
      i. Agenda pending previous meeting.
   c. Thursday, 2015/04/09 @ 11 AM CST
      i. Agenda pending previous meeting.
   d. At the April 9th meeting, we will determine if there is enough interest to continue the meetings.
3. Github issues list (insert link here)

Glossary of Terms:

Database modeling is a discipline that is rife with its own terms and jargon. The authors of this document have tried to be consistent with the use of these terms as defined below in the glossary. If the readers of this document believe that these terms are being used inconsistently please bring this to the attention of the authors.

Table 1: Glossary of Terms

Term	Definition	Notes	Reference
Attribute	The adjectives that describe a 'noun' (that is, an entity)	We tried to use the terms entity and attribute only at the DAM, Conceptual, and Logical level.	http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html
BRIDG	Biomedical Research Integrated Domain Group	CIBMTR has an association with the BRIDG SCC group to understand the content within the BRIDG model. We look to the BRIDG SCC	www.bridgmodel.org

Term	Definition	Notes	Reference
		to represent the interests of the BRIDG consortium	
BRIDG HCT	The project to incorporate data from transplant Centers to be shared with the CIBMTR into the BRIDG model.	This is a project team consisting of members of CIBMTR, NMDP, BRIDG SCC, and MD Anderson	
BRIDG HCT 1.0	The physical model and supporting documents that describe the output of the BRIDG HCT team.		
Camel Case	A convention in database modeling and many computer programming languages to define a unit with a base word and qualifiers such that the reader can distinguish separate words by the use of mixed upper and lower case. (The Capital Letters are the 'hump in the camel')	An example used in the DAM, the Conceptual, and Logical Model: PerformedProcedure. This denotes that 'Performed' is a qualifier on the word 'Procedure' without having to insert the space. Supported by many RDBMS platforms, but not supported by ANSI SQL. Because of this, we switch out of Camel Case when we switch to the Physical Model.	
CDE	Common Data Element	Term used to describe a unit of data. The caDSR (Cancer Data Standards Repository) is a powerful repository for storing definitions on units of data within the context (domain) of the National Cancer Institute	https://wiki.nci.nih.gov/display/caDSR/caDSR+Wiki
Column	The units of storage within a Table	We tried to use the term 'table' and 'column' only at the physical level.	
DDL	Data Definition Language	A Database management term that describes the syntax used to create tables and columns within an RDBMS. In contrast, DML (Data Modifying Language) is about insert, deleting, and updating data within tables and columns.	
Data type	Restriction on the kind of data a particular column can hold. If a column is defined as 'Number', then one	Constraining columns on datatypes is a powerful device for making data within a database consistent. If you are trying to store a value such as	

Term	Definition	Notes	Reference
	cannot store character data in this column.	'Number of Transplant', and you allow the user to store the word 'apple', this is handled as a data quality problem that could have been avoided.	
De-normalize	A verb that describes the practice of determining that an attribute or attributes that could be described at a higher level of abstraction are more convenient to place in a descendent entity or entities.	This is a term used mostly in data warehousing. It is the practice of knowingly violating Boyce-Codd 3rd Normalization, but due to clarity or performance reasons, it is determined the benefits outweigh the risks	
Domain	A collection of abstract concepts that have some connection. The collection of entities used for study-driven protocol is the BRIDG domain. The collection of entities for defining Hematopoietic Cell transplantation is the HCT Domain. The entities used to define HCT laboratory tests within the BRIDG model is the 'PerformedObservation Result' domain	This is a heavily overloaded term within the database modeling community.	
Entity	An abstract definition of a 'noun' that describes a discrete set of data.	We tried to use the terms entity and attribute only at the Domain Analysis Model (DAM), Conceptual, and Logical level.	http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html
GUID	Globally Unique Identifier		
HCT	Hematopoietic Cell Transplantation		
HL7	Health Level 7.	"Founded in 1987, Health Level Seven International (HL7) is a not-for-profit, ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health	http://www.hl7.org/

Term	Definition	Notes	Reference
		information that supports clinical practice and the management, delivery and evaluation of health services" -- http://www.hl7.org/about/index.cfm?ref=nav	
Model	In this context, it means a pictorial or representational presentation of a Domain. The Model of a Conceptual Domain would have less detail than a logical or physical model	Consistent with 'Domain', this is a heavily over-loaded term.	
OID	Unique Object Identifier	Nomenclature for managing values required for global uniqueness	http://www.hl7.org/implement/standards/product_brief.cfm?product_id=210
RDBMS	Relational Database Management System.	Examples include open-source solutions such as mySQL, and vendor solutions such as Microsoft, Oracle, etc.	http://www.databasedir.com/what-is-rdbms/
SCC	Semantic Coordinating Committee	Domain modelers tasked with maintaining and enhancing the BRIDG model. Subject-matter-experts for teams (Such as the BRIDG HCT team) who wish to model their domain's content in the BRIDG model	www.bridgmodel.org
Table	A unit of storage within an RDBMS consisting of rows and columns.	We tried to use the term 'table' and 'column' only at the physical level.	
UML	Unified Modeling Language	Nomenclature used within the BRIDG model	http://www.uml.org/

BRIDG HCT

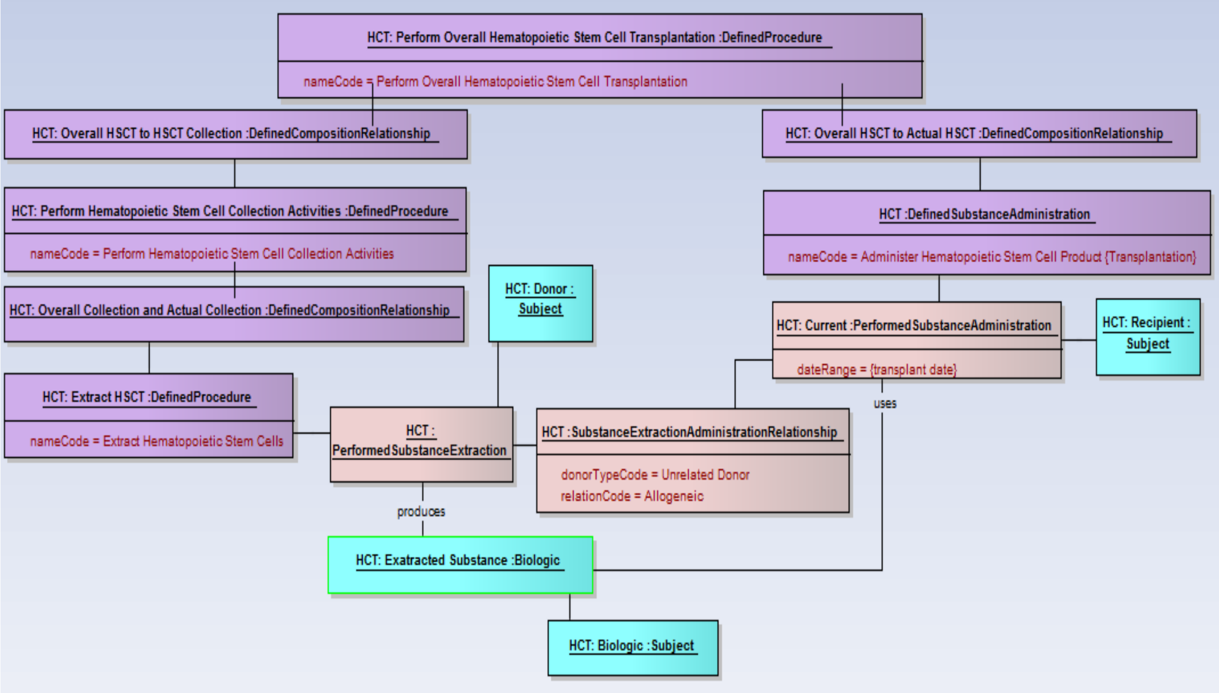
The BRIDG HCT Project consisted of a team of modelers from the CIBMTR, NMDP, and MD Anderson, with assistance from the BRIDC SCC team. The output of their work were multiple artifacts culminating in the DDL (Data Definition Language) sufficient to create a physical database that would contain the data specified in the mapping path.

The Core of the model that describes the HCT domain is encapsulated in Figure1. Every CDE can be tied back to this diagram, because every CDE ties back to a Subject of a particular type. All Subjects are entities who have activities performed upon them, or have actions performed on their behalf.

Table 2: The Three Kinds of Subjects in BRIDG HCT:

HCT Subject	**Definition**	**Notes**
Recipients	A recipient is the subject who receives hematopoietic cells, or the person who is the focus of a clinical trial or other focus of study	
Donors	Donor is defined in the broadest sense, in that this subject is the source of the hematopoietic cell product. Therefore, unrelated donors, family members, the patient, and cord blood units are all considered to be donors, as they are the provenance of the hematopoietic cell product	
product	The hematopoietic stem cell product itself. Anything that refers to a material, product, or biologic refers to the product as the subject.	In BRIDG, The 'Product' that is extracted is a 'Biologic'. The ancestor of 'Biologic' and 'Product' is 'Material'. Note the descendent of 'Product' ('Drug') is called out here because some HCT products (namely, cord blood units), are considered to be a drug by the FDA

Figure 1: The three kinds of Subjects and how they link together in a Transplantation:


HCT: Perform Overall Hematopoietic Stem Cell Transplantation :DefinedProcedure
nameCode = Perform Overall Hematopoietic Stem Cell Transplantation
HCT: Overall HSCT to HSCT Collection :DefinedCompositionRelationship
HCT: Overall HSCT to Actual HSCT :DefinedCompositionRelationship
HCT: Perform Hematopoietic Stem Cell Collection Activities :DefinedProcedure
nameCode = Perform Hematopoietic Stem Cell Collection Activities
HCT :DefinedSubstanceAdministration
nameCode = Administer Hematopoietic Stem Cell Product {Transplantation}
HCT: Overall Collection and Actual Collection :DefinedCompositionRelationship
HCT: Donor : Subject
HCT: Current :PerformedSubstanceAdministration
dateRange = {transplant date}
HCT: Recipient : Subject
uses
HCT: Extract HSCT :DefinedProcedure
nameCode = Extract Hematopoietic Stem Cells
HCT : PerformedSubstanceExtraction
HCT :SubstanceExtractionAdministrationRelationship
donorTypeCode = Unrelated Donor
relationCode = Allogeneic
produces
HCT: Exatracted Substance :Biologic
HCT: Biologic :Subject

Modeling Levels

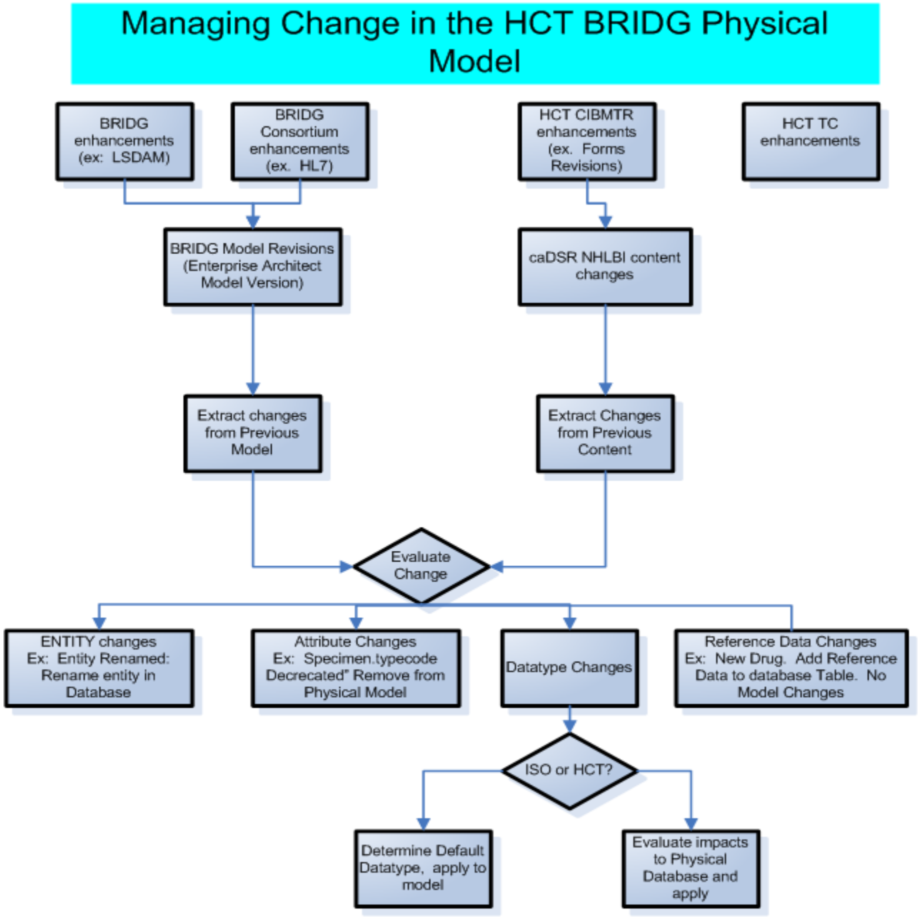
One of the first design decisions that was made for the BRIDG HCT 1.0 physical model, was to break the project into separate components. This exercise is a standard modeling exercise of first defining the conceptual layer, a logical layer, and the physical layer.

We chose to break the modeling effort into these component pieces in order to make it easier to revisit each of these pieces when we incorporated new content into the BRIDG model. If the BRIDG HCT 1.0 model were a one-time exercise, then we could have made the decision to go directly from the domain analysis model to a physical model. However, we know that the BRIDG model is a living document, and the domain of hematopoietic cell transplantation is also a growing field. We made the choice to break the modeling effort into separate pieces in order to facilitate change.

Facilitating Change

Note the different avenues of change that must be incorporated within the BRIDG HCT project to make the model extensible.

Figure 2: Defining avenues of modifications to BRIDG HCT Model


Managing Change in the HCT BRIDG Physical Model
BRIDG enhancements (ex: LSDAM)
BRIDG Consortium enhancements (ex. HL7)
HCT CIBMTR enhancements (ex. Forms Revisions)
HCT TC enhancements
BRIDG Model Revisions (Enterprise Architect Model Version)
caDSR NHLBI content changes
Extract changes from Previous Model
Extract Changes from Previous Content
Evaluate Change
ENTITY changes Ex: Entity Renamed: Rename entity in Database
Attribute Changes Ex: Specimen.typecode Decrecated" Remove from Physical Model
Datatype Changes
Reference Data Changes Ex: New Drug. Add Reference Data to database Table. No Model Changes
ISO or HCT?
Determine Default Datatype, apply to model
Evaluate impacts to Physical Database and apply

Data Governance

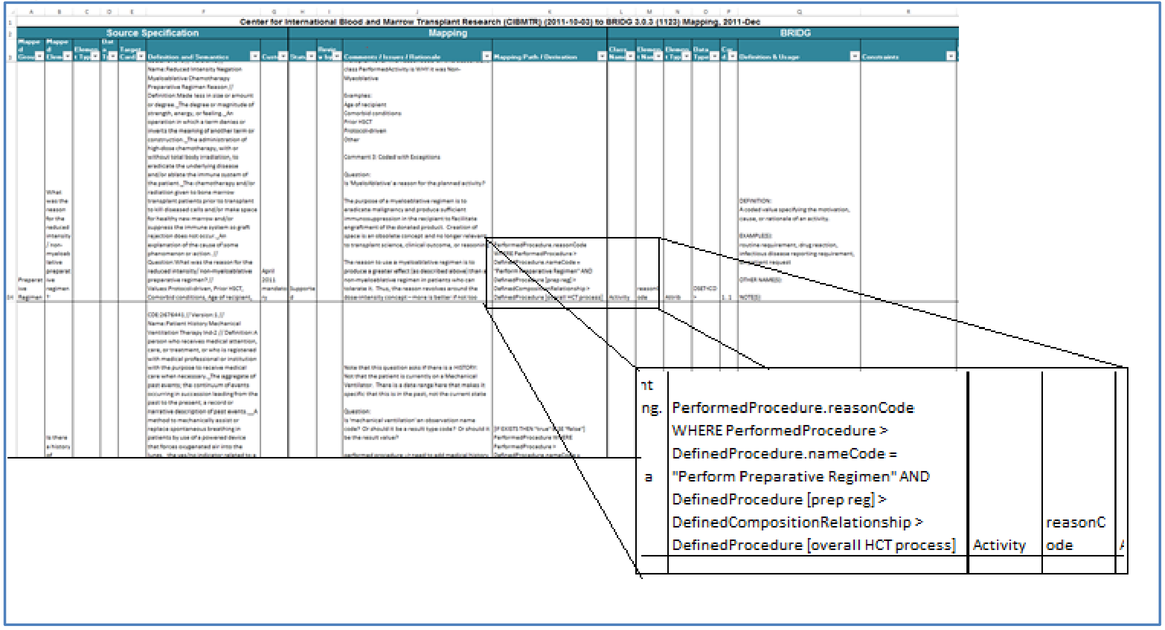
Part of this modeling exercise was also the to discuss the issue of data governance. The details of data governance are out of scope of this document. They will be accessible on the github site referred to above.

Conceptual Level:

The conceptual layer was used to determine the entities necessary for mapping all of the CDE's that have been identified to us by the data governance team of the CIBMTR. The mapping spreadsheet on the BRIDG model website (www.bridgmodel.org) defines the target entity. But it does not explicitly state all of the entities that are used in order to arrive at the target attributes. That is done by parsing the cells within the Excel spreadsheet.

In the example, the target entity and attribute are ‘Activity’ and ‘reasonCode’. The mapping path references ‘PerformedProcedure’, ‘DefinedProcedure’, and ‘DefinedCompositionRelationship’ to fully describe this CDE.

Figure 3:



Center for International Blood and Marrow Transplant Research (CIBMTR) (2011-10-03) to BRIDG 3.0.3 (1122) Mapping, 2011-Dec

Source Specification | Mapping | BRIDG

nt			
ng.	PerformedProcedure.reasonCode WHERE PerformedProcedure > DefinedProcedure.nameCode = "Perform Preparative Regimen" AND DefinedProcedure [prep reg] > DefinedCompositionRelationship > DefinedProcedure [overall HCT process]	Activity	reasonCode
a			

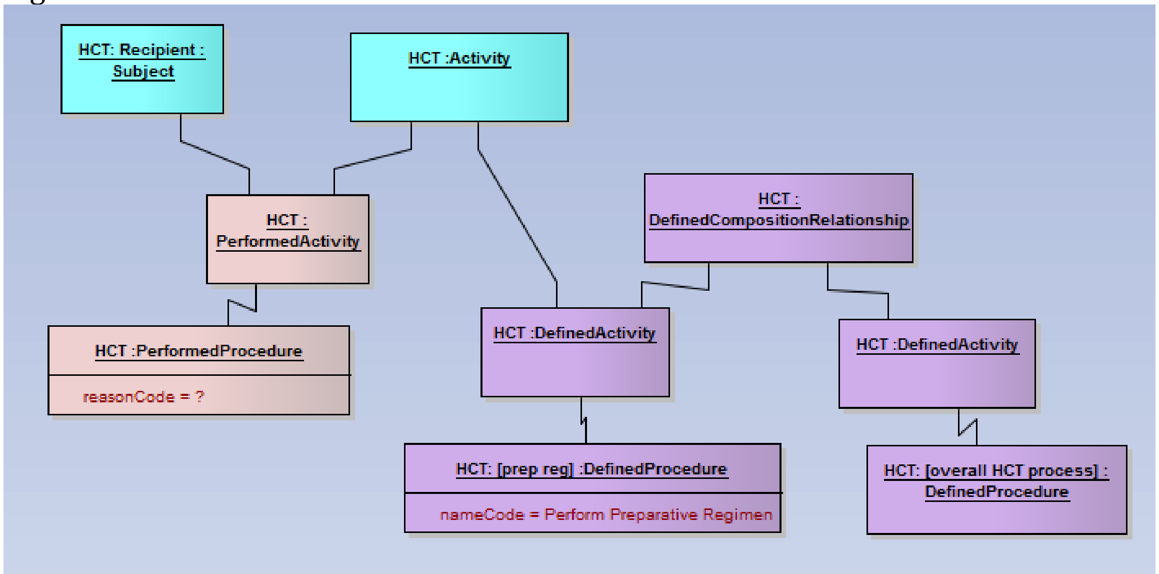
Also, the mapping paths do not list all necessary attributes. This was done in the interest of space during the actual mapping exercise that was finished in 2012. The conceptual model was used as a verification that indeed all of the entities that we needed plus the necessary ancestors of those entities were in one place so that we could perform a full analysis. For example, the BRIDG model uses the subject entity at its core. All activities are in relationship to the subject. In the interests of space due to the limitations of the Excel version that we were using, we eliminated explicit references to the subject entity in many entries. However there are many necessary relationships that are linked through the subject entity, so in the conceptual layer we decided to explicitly to bring subject back as the core entity. Note that this can cause some confusion, as the entities that are listed in an instance diagram will include the subject, while the mapping path may not.

In the mapping spreadsheet for CDE 2737040 (What was the reason for the reduced intensity/ non-myeloablative preparative regimen?), the mapping path is specified as

> PerformedProcedure.reasonCode WHERE PerformedProcedure > DefinedProcedure.nameCode = "Perform Preparative Regimen" AND DefinedProcedure [prep reg] > DefinedCompositionRelationship > DefinedProcedure [overall HCT process]

While the instance diagram details the actual mapping path of 'Subject > Activity > Performed Activity > PerformedProcedure'.

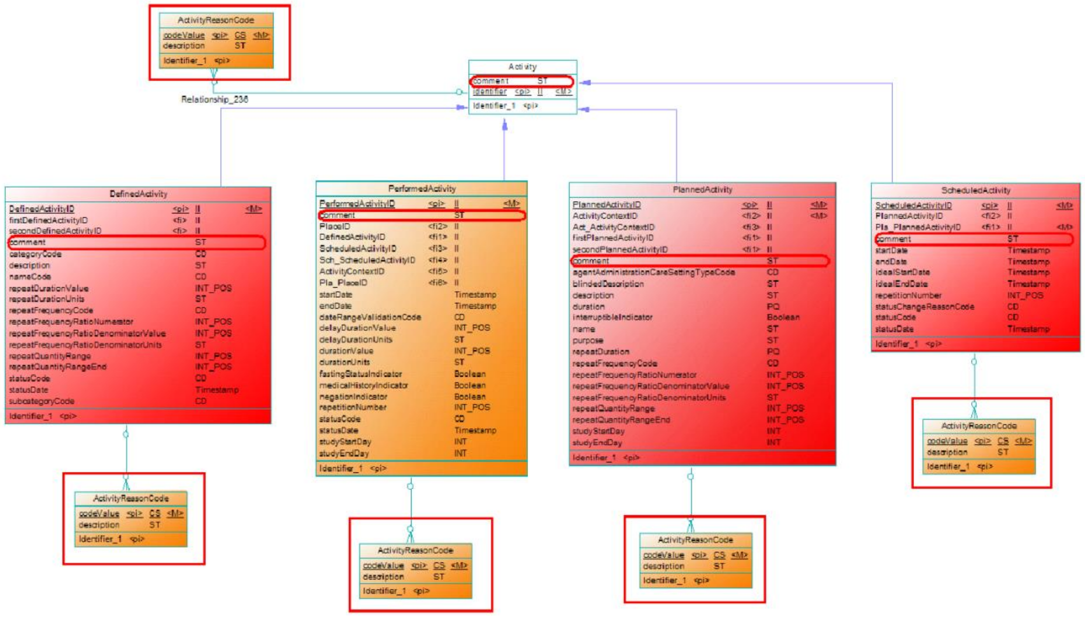
Figure 4:


HCT: Recipient : Subject
HCT :Activity
HCT : PerformedActivity
HCT : DefinedCompositionRelationship
HCT :PerformedProcedure
reasonCode = ?
HCT :DefinedActivity
HCT :DefinedActivity
HCT: [prep reg] :DefinedProcedure
nameCode = Perform Preparative Regimen
HCT: [overall HCT process] : DefinedProcedure

ACTIVITY Denormalization

We decided to denormalize the attributes on the activity entity into its direct descendants. There was no CDE that was only at the activity level. Therefore, the modelers felt there was very little utility in maintaining this entity through the descendent levels.

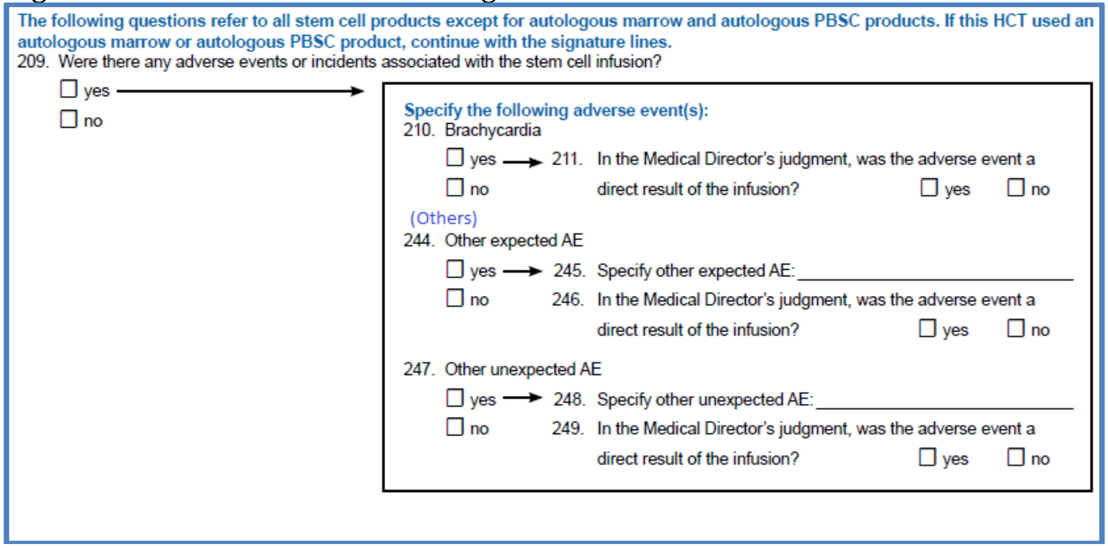
Figure 5. ‘Activity’ Denormalization



Adverse event

Here is an example of how a concept can be described at the domain analysis level, but cannot be easily implemented due to constraints of the particular domain. The CIBMTR collects adverse events as defined in Figure 6:

Figure 6 From Form 2006 collecting Adverse Event Information:



The following questions refer to all stem cell products except for autologous marrow and autologous PBSC products. If this HCT used an autologous marrow or autologous PBSC product, continue with the signature lines.

209. Were there any adverse events or incidents associated with the stem cell infusion?

□ yes →
□ no

Specify the following adverse event(s):

210. Brachycardia

□ yes → 211. In the Medical Director's judgment, was the adverse event a
□ no direct result of the infusion? □ yes □ no

(Others)

244. Other expected AE

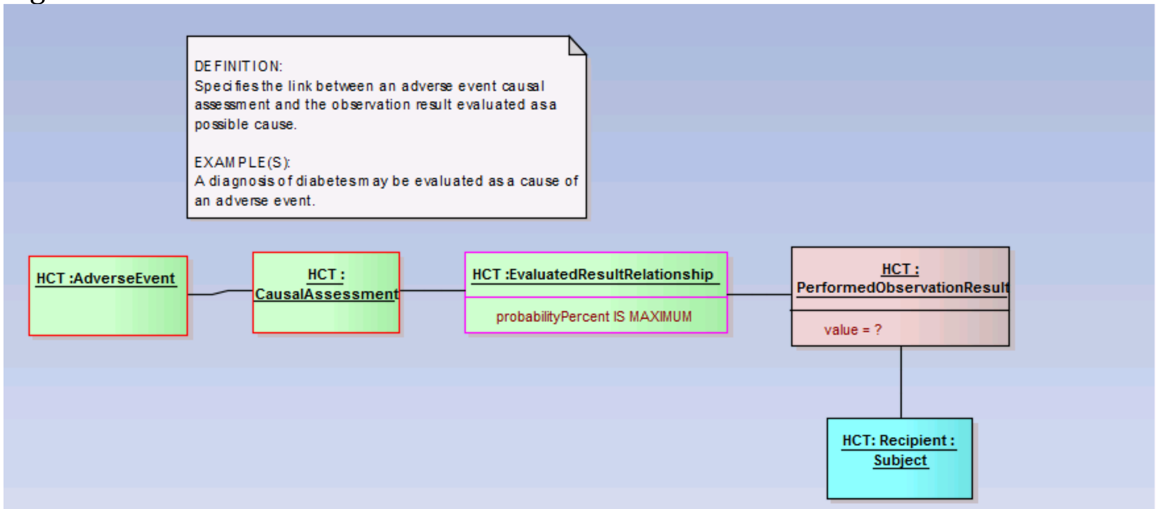
□ yes → 245. Specify other expected AE: ______________________
□ no 246. In the Medical Director's judgment, was the adverse event a
direct result of the infusion? □ yes □ no

247. Other unexpected AE

□ yes → 248. Specify other unexpected AE: ______________________
□ no 249. In the Medical Director's judgment, was the adverse event a
direct result of the infusion? □ yes □ no

This data must be transmittable between a transplant center and the CIBMTR or between transplant centers. Is not always possible to know if the adverse event causation was due to an 'Observation' or an 'Action'. Note that even if each sub-question could be assigned to either 'Action' or 'Observation', that the inclusion of the 'Other Specify' option re-introduces the ambiguity.

Figure 7: 'AdverseEvent' Due to an 'Observation':

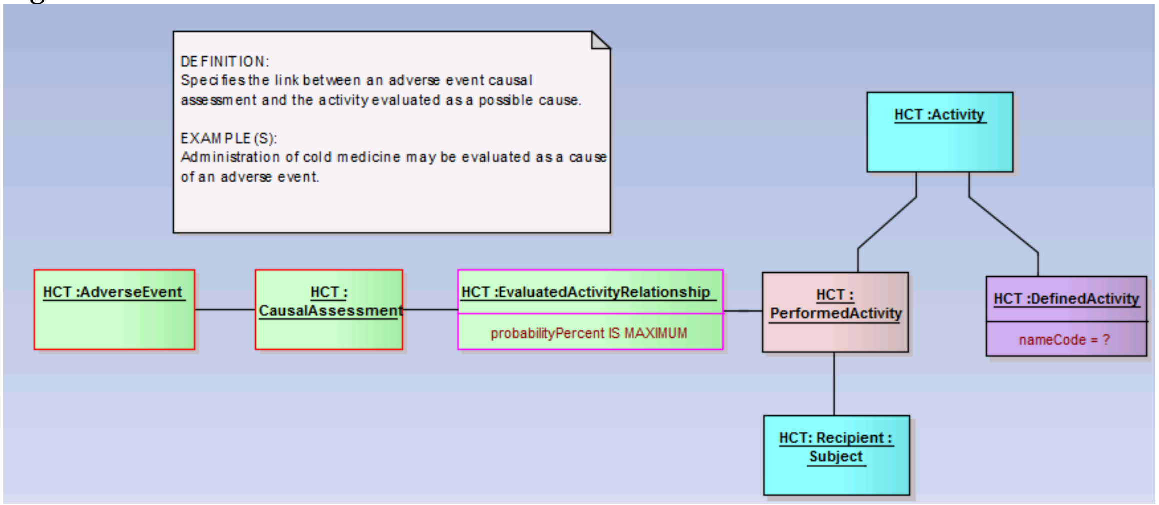


DEFINITION:
Specifies the link between an adverse event causal assessment and the observation result evaluated as a possible cause.

EXAMPLE(S):
A diagnosis of diabetes may be evaluated as a cause of an adverse event.

HCT :AdverseEvent
HCT : CausalAssessment
HCT :EvaluatedResultRelationship
probabilityPercent IS MAXIMUM
HCT : PerformedObservationResult
value = ?
HCT: Recipient : Subject

Figure 8: 'AdverseEvent' Due to an 'Action':

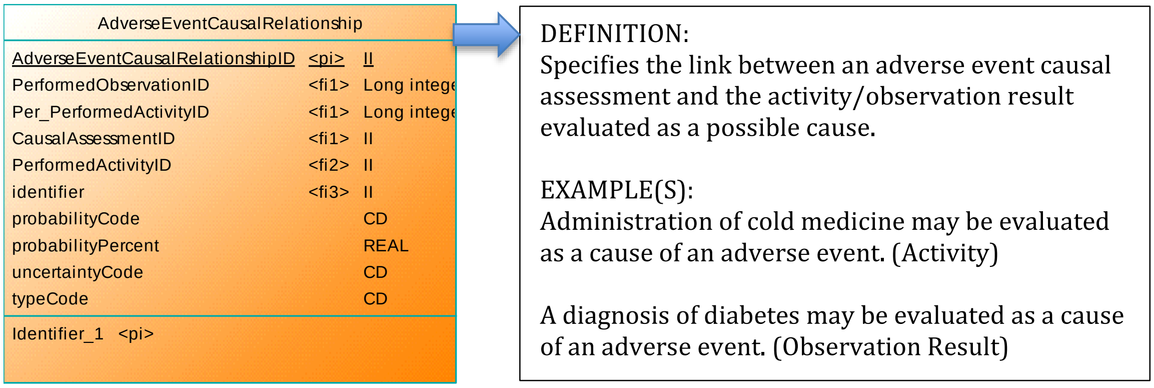


DEFINITION:
Specifies the link between an adverse event causal assessment and the activity evaluated as a possible cause.

EXAMPLE(S):
Administration of cold medicine may be evaluated as a cause of an adverse event.

In fact, it could be the intent of a study to determine the causal relationship of an adverse event. Because we did not wish to force a transplant center to make a decision upon the causal relationship of an adverse event for perhaps arbitrary reasons, we created a new ancestor for these two kinds of adverse events. The BRIDG HCT modelers intend to bring this to the attention of the BRIDG SCC to determine if they wish to include this in the larger BRIDG model or not. The BRIDG HCT modelers intend to maintain this structure in the BRIDG HCT domain.

Figure 9: New Logical Relationship to describe unknown causation of adverse events.



AdverseEventCausalRelationship

AdverseEventCausalRelationshipID	<pi>	II
PerformedObservationID	<fi1>	Long intege
Per_PerformedActivityID	<fi1>	Long intege
CausalAssessmentID	<fi1>	II
PerformedActivityID	<fi2>	II
identifier	<fi3>	II
probabilityCode		CD
probabilityPercent		REAL
uncertaintyCode		CD
typeCode		CD

Identifier_1 <pi>

DEFINITION:
Specifies the link between an adverse event causal assessment and the activity/observation result evaluated as a possible cause.

EXAMPLE(S):
Administration of cold medicine may be evaluated as a cause of an adverse event. (Activity)

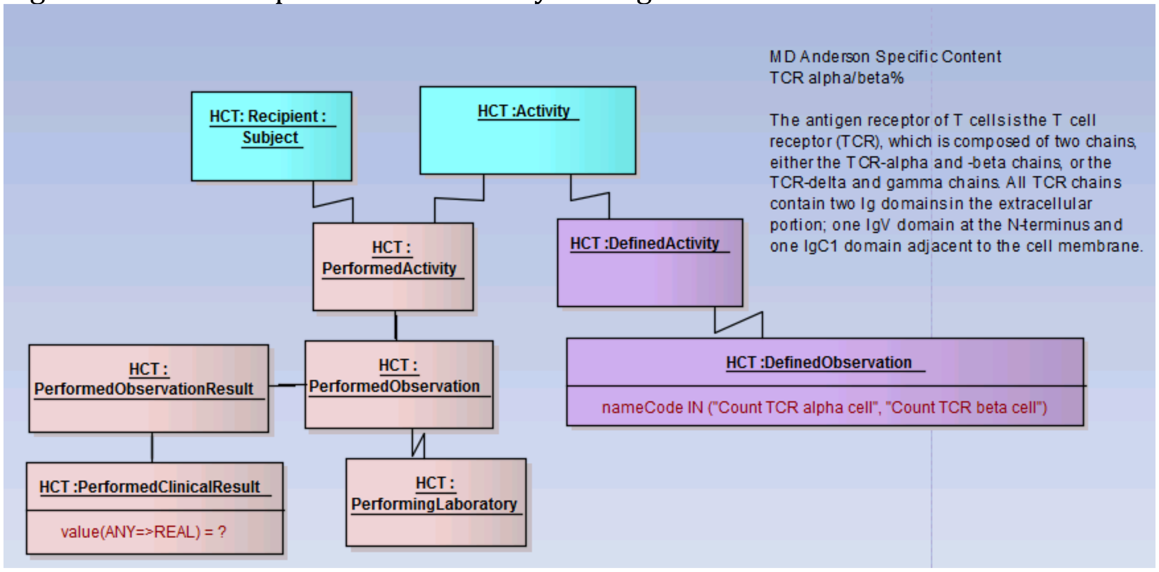
A diagnosis of diabetes may be evaluated as a cause of an adverse event. (Observation Result)

The conceptual model is actually in two pieces.   This is an artificial distinction, and is due mainly to the use of the tools that we are using to manage the model. The

BRIDG model is contained in an Enterprise Architect diagram, and the CIBMTR uses Powerdesigner to model its internal artifacts. Either of these tools could be used to manage the entire suite of artifacts, and this is a distinction used by CIBMTR modelers to comply with internal organization policies. Any organization using any of the artifacts provided by these projects should use their own organizational modeling tools. There are conversion tools that can be used to move artifacts from one suite of tools to another, and indeed that is what we used to convert this first half of the conceptual model to the second half the conceptual model. Those instructions are left to each individual organization.

The first half of the conceptual is a copy of the BRIDG 3.2 domain model with the addition of instance diagrams for every CDE that is listed in the mapping spreadsheet document included in the BRIDG 3.2 HCTv1.0 2012 release . Note that in the BRIDG 3.2 release documents there are some entries the mapping documents did not have, and there are some CDEs that were not mapped. In the first case, these are examples of contents that have no CDEs, but were useful to our collaborator. These were included in the 2012 BRIDG 3.2 release to establish that it was possible for collaborators to map their content that may not necessarily be associated to a CDE:

Figure 10: Center-Specific content not yet assigned to a CDE:

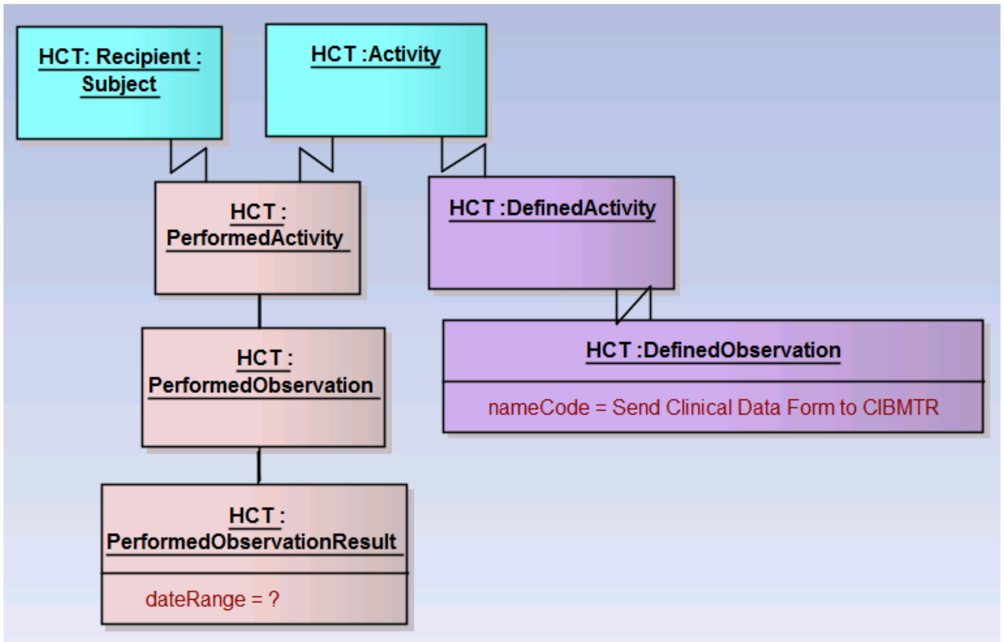

HCT: Recipient : Subject
HCT :Activity
HCT : PerformedActivity
HCT :DefinedActivity
HCT : PerformedObservationResult
HCT : PerformedObservation
HCT :DefinedObservation
nameCode IN ("Count TCR alpha cell", "Count TCR beta cell")
HCT :PerformedClinicalResult
value(ANY=>REAL) = ?
HCT : PerformingLaboratory

MD Anderson Specific Content
TCR alpha/beta%

The antigen receptor of T cellsisthe T cell receptor (TCR), which is composed of two chains, either the TCR-alpha and -beta chains, or the TCR-delta and gamma chains. All TCR chains contain two Ig domainsin the extracellular portion; one IgV domain at the N-terminus and one IgC1 domain adjacent to the cell membrane.

Also, there were some CDEs that facilitated the collection mechanism of the content. This is outside the scope of BRIDG, but the BRIDG model can contain the information in the 'PerformedObservation>PerformedObservationResult' construction.

Figure 11: CDE 2866943 Clinical Data Form Date Received



Note that BRIDG is flexible enough to contain both these kinds of content. In the interest of completeness for the physical model, all of these entries were mapped to the conceptual model.

The creation of an instance diagram for every CDE was not a trivial effort, but it was done in order to take advantage of some of the tools that are contained inside Enterprise Architect. If one creates an instance diagram with every object needed to describe that CDE, one can extract a list of all of the entities needed to support that CDE. This was used to determine that we had the complete set of all entities from the BRIDG 3.2 model in order to convert it to our next level of modeling.

This was useful in the initial exercise of extracting the needed entities for the next level of modeling, but we intend this to be a valuable resource when we incorporate change into the model. We now have a base set of objects that we know are necessary to HCT content, and we now can use that as a set in order to determine what changes are necessary for subsequent revisions of the model.

Notes that the instance diagrams as they are listed in the Enterprise Architect model contain little metadata. This is due solely to time constraints, and we intend to enrich these models with the associated questions, entity definitions, etc. to make them more comprehensible for external readers.

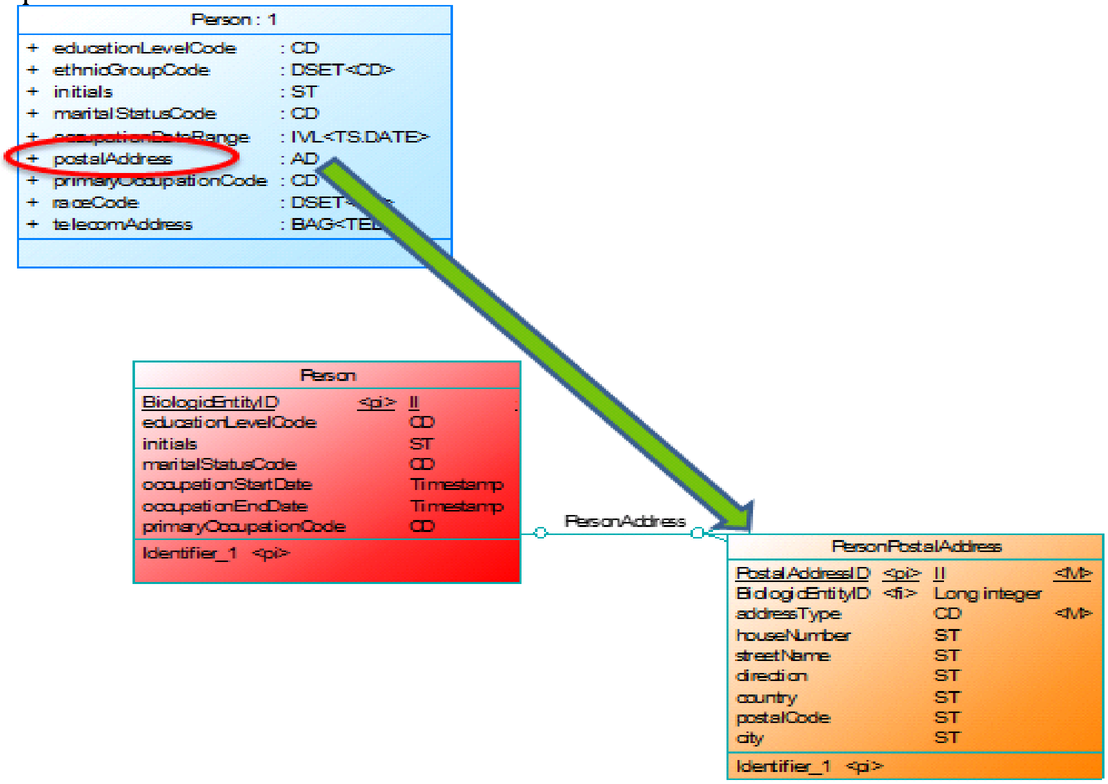
Another important decision we made at the conceptual level was to include only those entities that were explicitly used in the CDEs (with the noted exceptions above) but to include every attribute for every entity.

This was due primarily to a limitation of the tools and a constraint on time. The tools used do not make it easy to filter out desired attributes from an entity and this would have required coding. This coding effort was not undertaken for the BRIDG HCT 1.0 effort. We felt that the inclusion of these attributes would not be too onerous for the modeling effort, because we were going to be applying consistent data types for all attributes, and all the attributes listed have well-defined data types in the BRIDG model.

Logical Level:
The logical level was used to break apart implicit one-to-many relationships that existed in the conceptual layer. A good example of this is the address data type that exists on the person entity. From a conceptual layer, it is sufficient to say that each person has an address. It is even appropriate to say that they have 1-to- many relationship (One person can have one-to-many addresses). It is at the logical layer that this relationship is broken down into the individual entities that are necessary to contain this concept. The address data type was broken into a person address entity, and the minimum attributes for an address were added to this entity. The minimum set of address attributes were those that were defined with the HL7 data type.

A full description of the HL7 datatypes can be found in 'Abstract data types' within the HL7 Normative Edition, R2 or HL7 Version 3 Standard: XML Implementation Technology Specification R2: ISO-Harmonized Data Types, Release, both accessible from the HL7 website (www.hl7.org)
1

Figure 12: An example of expanding a Conceptual Attribute to a logical representation



When the one-to-many, many-to-many, and complex data type relationships have been defined at the logical level it is now possible to take the model to a physical level.

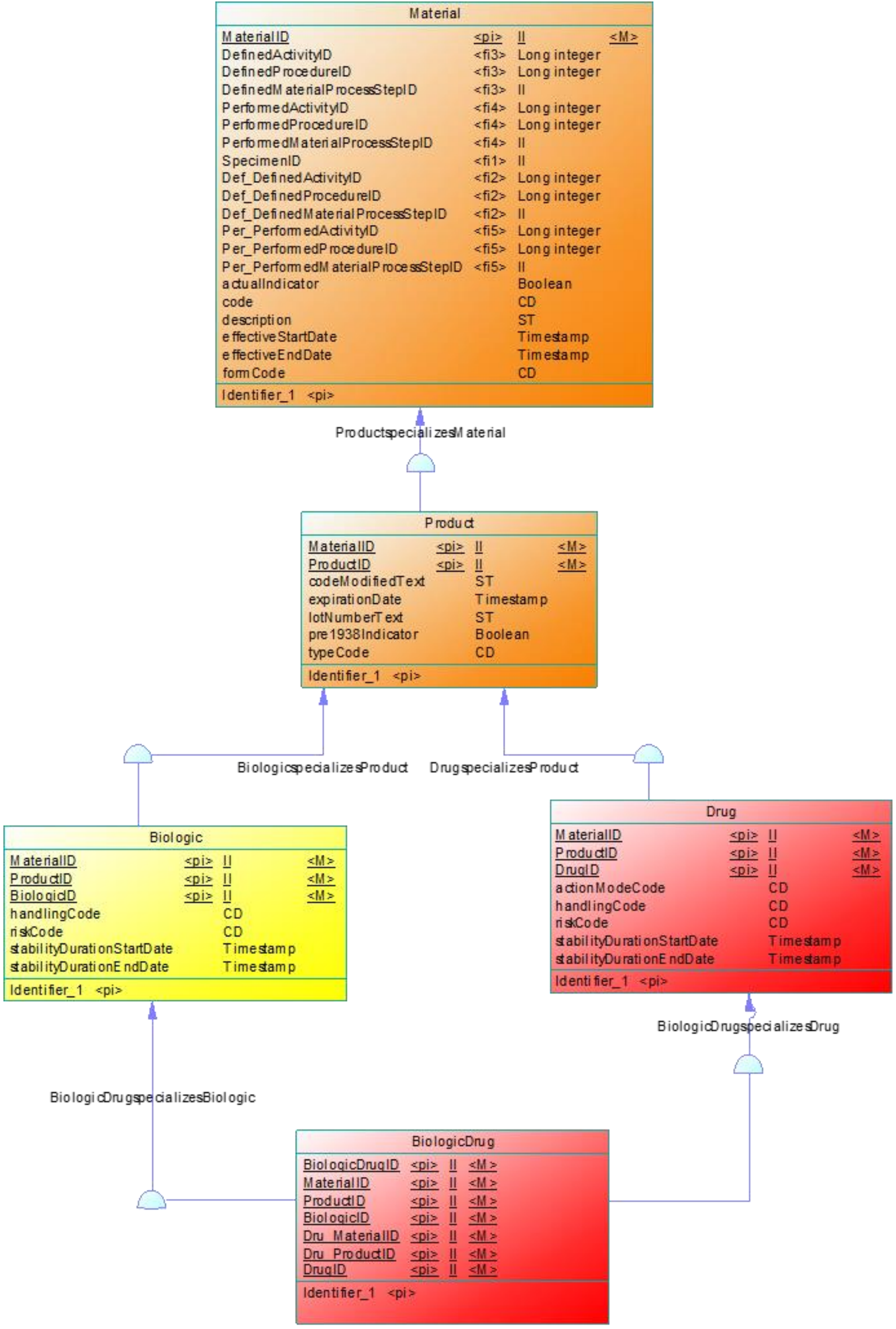
'BiologicDrug'

One of the more difficult concepts to model using the BRIDG model is the distinction between 'Biologics' and 'Drugs'. This is due to the inherent nature of the BRIDG model and the HCT domain. Not all drugs are 'Biologics', and not all 'Biologic's are 'Drugs', but for the BRIDG HCT domain, they are. Note also that cord blood is regulated as a drug, while at the time of this writing, bone marrow and PBSC (peripheral blood stem cells) are not. The BRIDG model is intended to serve more than the HCT community, and therefore we (the BRIDG HCT modelers) recognize the distinction between these entities as necessary.

Note the double inheritance of 'BiologicDrug'. The BRIDG model itself is designed with only one parent (generalization) for an entity. We accept the risk that the BRIDG consortium may decide to modify the structure and we will need to maintain the introduction of this entity that describes our content more fully.

We chose to do this at the logical level because the authors felt managing change at the conceptual level would be easier, and because of the double-inheritance complication, this new entity should be isolated from those possible changes if this were applied at the logical level.

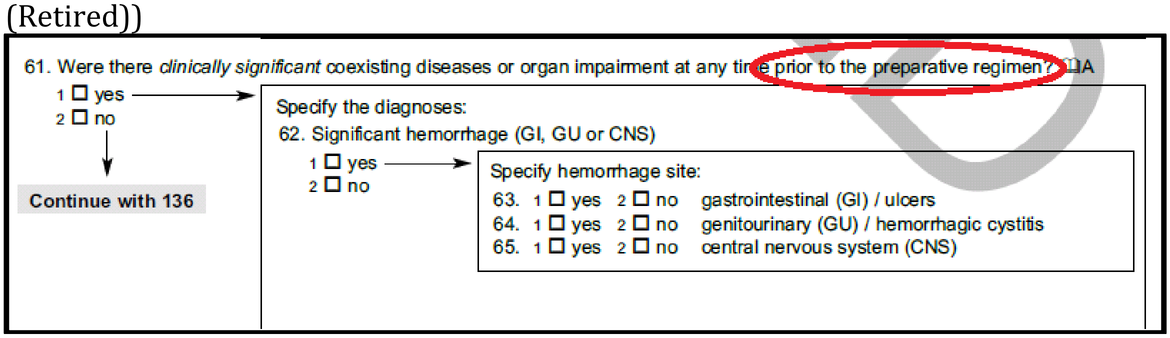
Figure 13: Double inheritance for ‘BiologicDrug’


Material
MaterialID <pi> II <M>
DefinedActivityID <fi3> Long integer
DefinedProcedureID <fi3> Long integer
DefinedMaterialProcessStepID <fi3> II
PerformedActivityID <fi4> Long integer
PerformedProcedureID <fi4> Long integer
PerformedMaterialProcessStepID <fi4> II
SpecimenID <fi1> II
Def_DefinedActivityID <fi2> Long integer
Def_DefinedProcedureID <fi2> Long integer
Def_DefinedMaterialProcessStepID <fi2> II
Per_PerformedActivityID <fi5> Long integer
Per_PerformedProcedureID <fi5> Long integer
Per_PerformedMaterialProcessStepID <fi5> II
actualIndicator Boolean
code CD
description ST
effectiveStartDate Timestamp
effectiveEndDate Timestamp
formCode CD
Identifier_1 <pi>
ProductspecializesMaterial
Product
MaterialID <pi> II <M>
ProductID <pi> II <M>
codeModifiedText ST
expirationDate Timestamp
lotNumberText ST
pre1938Indicator Boolean
typeCode CD
Identifier_1 <pi>
BiologicspecializesProduct
DrugspecializesProduct
Biologic
MaterialID <pi> II <M>
ProductID <pi> II <M>
BiologicID <pi> II <M>
handlingCode CD
riskCode CD
stabilityDurationStartDate Timestamp
stabilityDurationEndDate Timestamp
Identifier_1 <pi>
Drug
MaterialID <pi> II <M>
ProductID <pi> II <M>
DrugID <pi> II <M>
actionModeCode CD
handlingCode CD
riskCode CD
stabilityDurationStartDate Timestamp
stabilityDurationEndDate Timestamp
Identifier_1 <pi>
BiologicDrugspecializesDrug
BiologicDrugspecializesBiologic
BiologicDrug
BiologicDrugID <pi> II <M>
MaterialID <pi> II <M>
ProductID <pi> II <M>
BiologicID <pi> II <M>
Dru_MaterialID <pi> II <M>
Dru_ProductID <pi> II <M>
DrugID <pi> II <M>
Identifier_1 <pi>

Time Points

Using the HL7 data types for the 'PerformedActivity.dateRange' implies that we know a start date and end date. Over the 30-year history of collecting data for the CIBMTR this date is not always known. Part of the forms-collection process used to the concept of 'time point.' This gave a strong understanding of when an activity occurred in the course of treatment in relation to other activities in the course of treatment but not necessarily a specific date.

Figure 14: An example of a 'time point' (From CIBMTR Form 2000, Version1 (Retired))



61. Were there *clinically significant* coexisting diseases or organ impairment at any time prior to the preparative regimen? A

1 □ yes → Specify the diagnoses:

2 □ no

Continue with 136

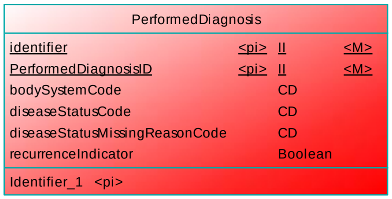
62. Significant hemorrhage (GI, GU or CNS)

1 □ yes → Specify hemorrhage site:

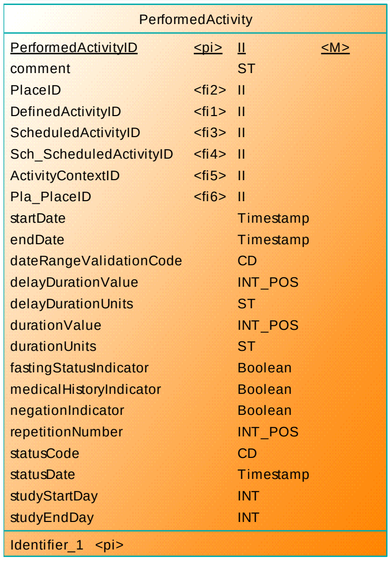
2 □ no

63.	1 □ yes	2 □ no	gastrointestinal (GI) / ulcers
64.	1 □ yes	2 □ no	genitourinary (GU) / hemorrhagic cystitis
65.	1 □ yes	2 □ no	central nervous system (CNS)

Note that in the form, the preparative regimen start date is noted, but not the date of diagnosis of a coexisting condition. So it would be inaccurate to have the end date for the 'PerformedActivity > Performed Diagnosis' be the same as the start date of the Preparative Regimen. Both of these are activities that are mapped to the 'PerformedActivity' entity.



PerformedDiagnosis			
identifier	<pi>	II	<M>
PerformedDiagnosisID	<pi>	II	<M>
bodySystemCode		CD	
diseaseStatusCode		CD	
diseaseStatusMissingReasonCode		CD	
recurrenceIndicator		Boolean	
Identifier_1 <pi>			

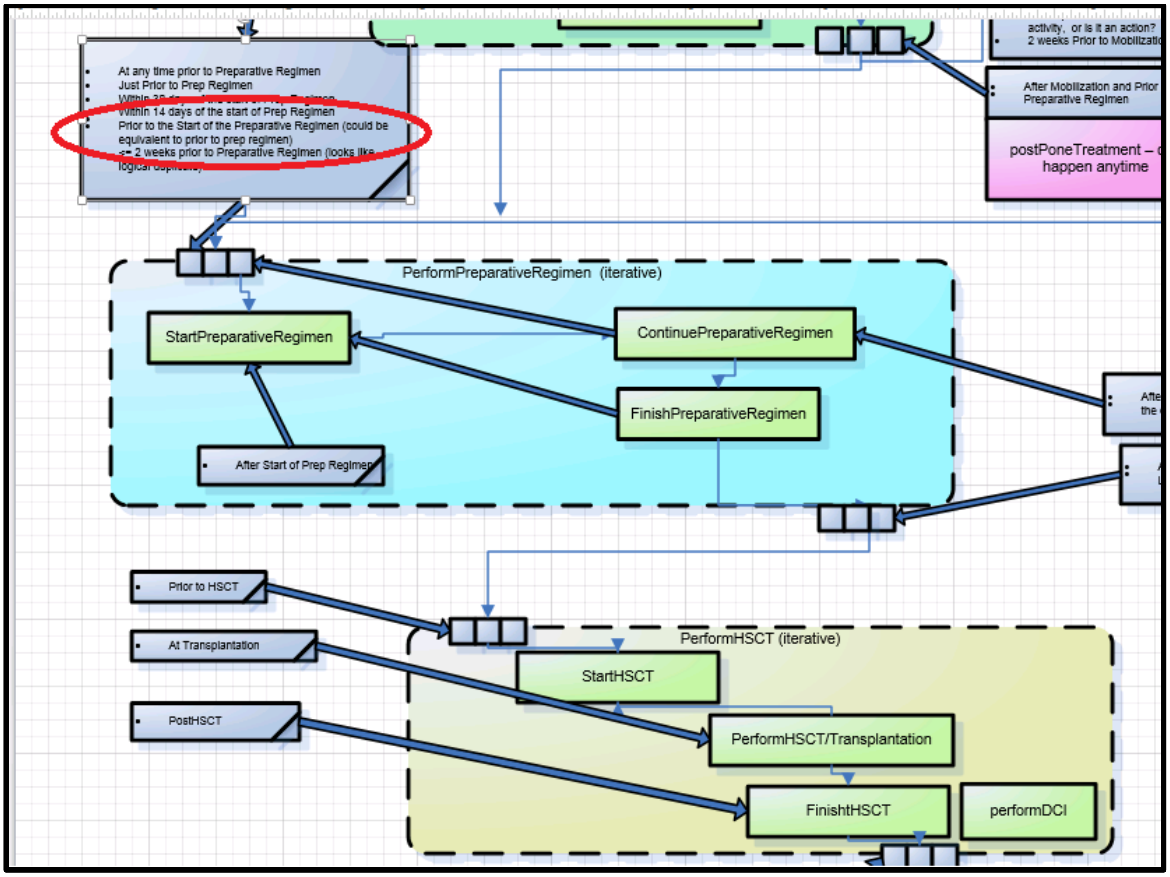


PerformedActivity			
PerformedActivityID	<pi>	II	<M>
comment		ST	
PlaceID	<fi2>	II	
DefinedActivityID	<fi1>	II	
ScheduledActivityID	<fi3>	II	
Sch_ScheduledActivityID	<fi4>	II	
ActivityContextID	<fi5>	II	
Pla_PlaceID	<fi6>	II	
startDate		Timestamp	
endDate		Timestamp	
dateRangeValidationCode		CD	
delayDurationValue		INT_POS	
delayDurationUnits		ST	
durationValue		INT_POS	
durationUnits		ST	
fastingStatusIndicator		Boolean	
medicalHistoryIndicator		Boolean	
negationIndicator		Boolean	
repetitionNumber		INT_POS	
statusCode		CD	
statusDate		Timestamp	
studyStartDay		INT	
studyEndDay		INT	
Identifier_1 <pi>			

There are many occurrences within the history of data collection by the CIBMTR where the relative order of events is highly known, but not the specific dates. We refer the reader to the activity diagram within the documentation set.

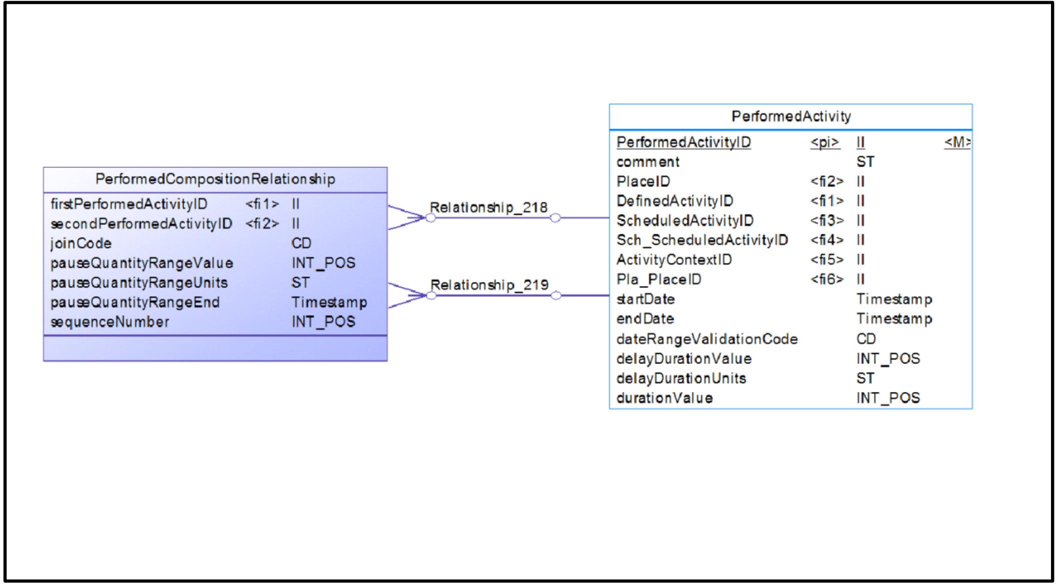
A snippet of the activity diagram is here, to denote how any activity that happens 'prior to the start of preparative regimen' relates to every other activity. Note that all the time-points on the forms are represented in the activity diagram.

Figure 15: demonstrating relationship of 'prior to preparative regimen' to other activities.


At any time prior to Preparative Regimen
Just Prior to Prep Regimen
Within 30 days of the start of Prep Regimen
Within 14 days of the start of Prep Regimen
Prior to the Start of the Preparative Regimen (could be equivalent to prior to prep regimen)
<= 2 weeks prior to Preparative Regimen (looks like logical duplicate)
activity, or is it an action?
2 weeks Prior to Mobilizatio
After Mobilization and Prior Preparative Regimen
postPoneTreatment – c happen anytime
PerformPreparativeRegimen (iterative)
StartPreparativeRegimen
ContinuePreparativeRegimen
FinishPreparativeRegimen
After Start of Prep Regimen
Afte the
Prior to HSCT
At Transplantation
PostHSCT
PerformHSCT (iterative)
StartHSCT
PerformHSCT/Transplantation
FinishtHSCT
performDCI

Within the BRIDG model, dates are used to define relative events. Because the physical data model requires an actual beginning date and end date, and this may not actually be known, we had to introduce the concept of 'Performed Composition Relationship'.

Figure 16: The creation of ‘PerformedCompositionRelationship’

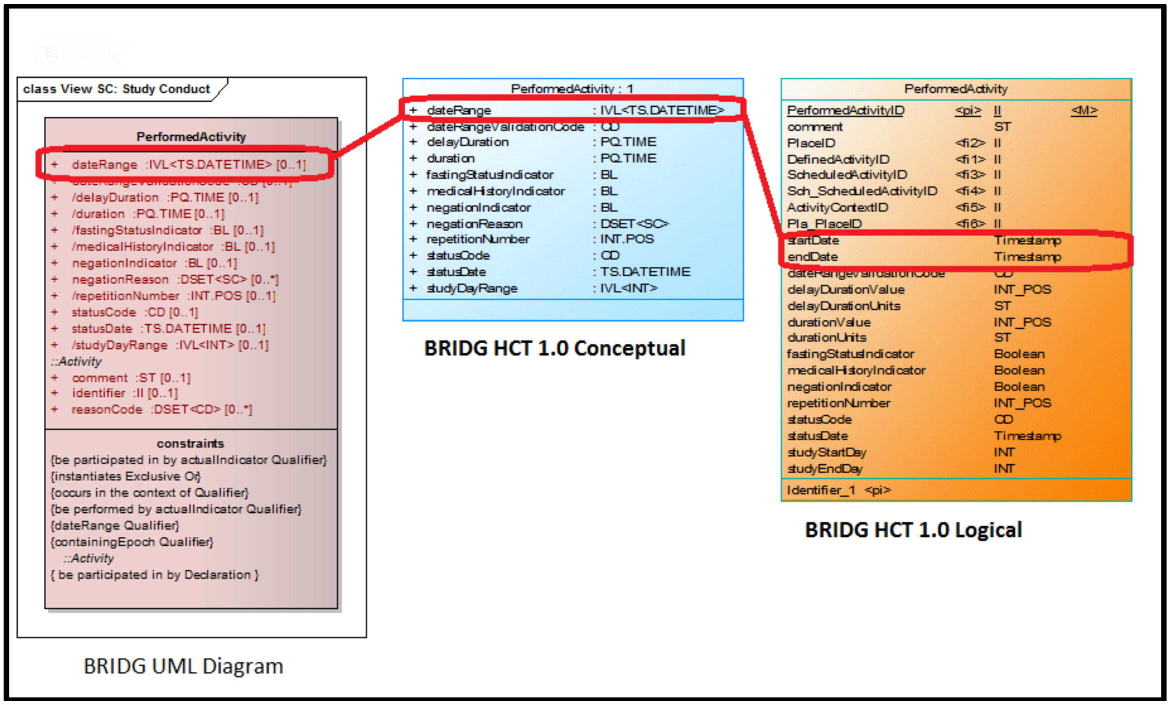

PerformedCompositionRelationship
firstPerformedActivityID <fi1> II
secondPerformedActivityID <fi2> II
joinCode CD
pauseQuantityRangeValue INT_POS
pauseQuantityRangeUnits ST
pauseQuantityRangeEnd Timestamp
sequenceNumber INT_POS
Relationship_218
Relationship_219
PerformedActivity
PerformedActivityID <pi> II <M>
comment ST
PlaceID <fi2> II
DefinedActivityID <fi1> II
ScheduledActivityID <fi3> II
Sch_ScheduledActivityID <fi4> II
ActivityContextID <fi5> II
Pla_PlaceID <fi6> II
startDate Timestamp
endDate Timestamp
dateRangeValidationCode CD
delayDurationValue INT_POS
delayDurationUnits ST
durationValue INT_POS

From the example above, we would create an entry in the ‘PerformedCompositionRelationship’ entity linking the ‘PerformedDiagnosis’ of a pre-existing composition, and the ‘PerformedActivity’ of starting the preparative regimen, where the relationship is ‘Before’. Note that now neither activity requires dates to define this chronological relationship.
Obviously if we have dates given to us for any of these activities these would override the ‘PerformedCompositionRelationship’ information. As more data is captured in EMR systems, this construction is less necessary.

We decided to include the definition of ‘PerformedCompositionRelationship’ at the logical level.   The BRIDG 3.2 model as defined by the BRIDG consortium adequately describes a date range.

We had also made the decision to apply HL7 data types to all attributes consistently to facilitate modifications to the model.   The conceptual model was designated as the place to deal with changes to the BRIDG model and the decomposition of complex data types. The ‘dateRange’ data type on ‘PerformedActivity’ is a time interval, and as such, a complex data type.   The BRIDG HCT modelers felt the logical model was the best place to deal with the consequences of those first decisions.

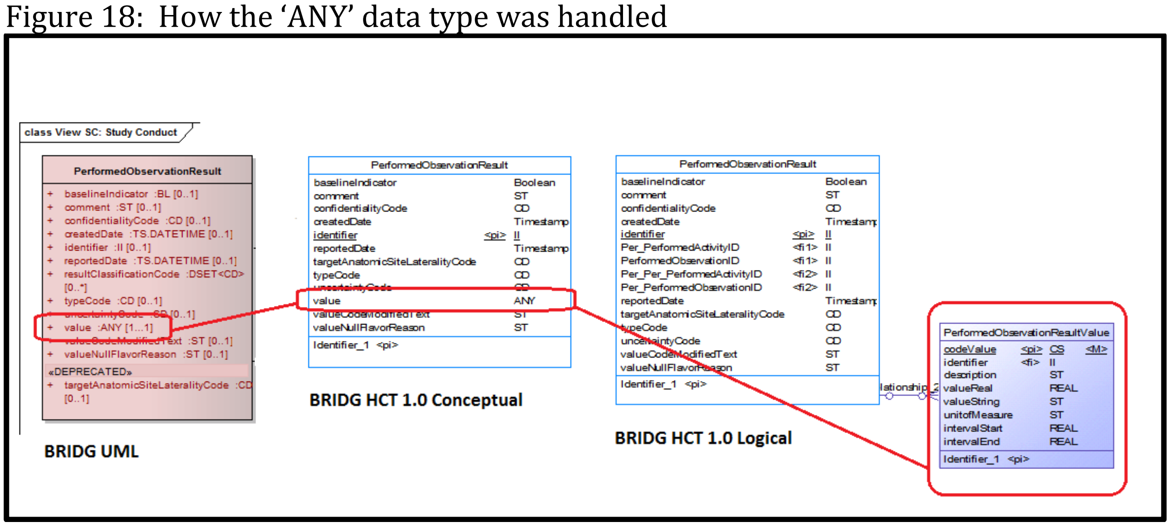
Figure 17: The transition of ‘dateRange’ from UML to Conceptual to Logical Models:


class View SC: Study Conduct
PerformedActivity
+ dateRange :IVL<TS.DATETIME> [0..1]
+ /delayDuration :PQ.TIME [0..1]
+ /duration :PQ.TIME [0..1]
+ /fastingStatusIndicator :BL [0..1]
+ /medicalHistoryIndicator :BL [0..1]
+ negationIndicator :BL [0..1]
+ negationReason :DSET<SC> [0..*]
+ /repetitionNumber :INT.POS [0..1]
+ statusCode :CD [0..1]
+ statusDate :TS.DATETIME [0..1]
+ /studyDayRange :IVL<INT> [0..1]
::Activity
+ comment :ST [0..1]
+ identifier :II [0..1]
+ reasonCode :DSET<CD> [0..*]
constraints
{be participated in by actualIndicator Qualifier}
{instantiates Exclusive Or}
{occurs in the context of Qualifier}
{be performed by actualIndicator Qualifier}
{dateRange Qualifier}
{containingEpoch Qualifier}
::Activity
{ be participated in by Declaration }
BRIDG UML Diagram
PerformedActivity : 1
+ dateRange : IVL<TS.DATETIME>
+ dateRangeValidationCode : CD
+ delayDuration : PQ.TIME
+ duration : PQ.TIME
+ fastingStatusIndicator : BL
+ medicalHistoryIndicator : BL
+ negationIndicator : BL
+ negationReason : DSET<SC>
+ repetitionNumber : INT.POS
+ statusCode : CD
+ statusDate : TS.DATETIME
+ studyDayRange : IVL<INT>
BRIDG HCT 1.0 Conceptual
PerformedActivity
PerformedActivityID <pi> II <M>
comment ST
PlaceID <fi2> II
DefinedActivityID <fi1> II
ScheduledActivityID <fi3> II
Sch_ScheduledActivityID <fi4> II
ActivityContextID <fi5> II
Pla_PlaceID <fi6> II
startDate Timestamp
endDate Timestamp
dateRangeValidationCode CD
delayDurationValue INT_POS
delayDurationUnits ST
durationValue INT_POS
durationUnits ST
fastingStatusIndicator Boolean
medicalHistoryIndicator Boolean
negationIndicator Boolean
repetitionNumber INT_POS
statusCode CD
statusDate Timestamp
studyStartDay INT
studyEndDay INT
Identifier_1 <pi>
BRIDG HCT 1.0 Logical

The ‘ANY’ Datatype

An application of using the BRIDG 3.2 model is the reference to the data type definition of ‘ANY’. At a certain point in time one has to make the decision about how this is going to be implemented in an RDBMS. For example the ‘PerformedObservationResult’ is a collection of laboratory results. These laboratory results can be a date, a character string, or any number. Note that the ‘ANY’ datatype could also incorporate an image of a laboratory report or some other kind of diagnostic image. The BRIDG HCT modeling team did not have to deal with these kinds of complications, because there are no CDEs that define a result of an observation in those terms. In other words, there are CDEs that define a conclusion or interpretation of an imaging test, but no CDE that encompasses the image itself. This limited the number of potential physical data types that the BRIDG HCT modeling team needed to consider.

Figure 18: How the ‘ANY’ data type was handled


class View SC: Study Conduct
PerformedObservationResult
+ baselineIndicator :BL [0..1]
+ comment :ST [0..1]
+ confidentialityCode :CD [0..1]
+ createdDate :TS.DATETIME [0..1]
+ identifier :II [0..1]
+ reportedDate :TS.DATETIME [0..1]
+ resultClassificationCode :DSET<CD> [0..*]
+ typeCode :CD [0..1]
+ uncertaintyCode :CD [0..1]
+ value :ANY [1...1]
+ valueCodeModifiedText :ST [0..1]
+ valueNullFlavorReason :ST [0..1]
«DEPRECATED»
+ targetAnatomicSiteLateralityCode :CD [0..1]
BRIDG UML
PerformedObservationResult
baselineIndicator Boolean
comment ST
confidentialityCode CD
createdDate Timestamp
identifier <pi> II
reportedDate Timestamp
targetAnatomicSiteLateralityCode CD
typeCode CD
uncertaintyCode CD
value ANY
valueCodeModifiedText ST
valueNullFlavorReason ST
Identifier_1 <pi>
BRIDG HCT 1.0 Conceptual
PerformedObservationResult
baselineIndicator Boolean
comment ST
confidentialityCode CD
createdDate Timestamp
identifier <pi> II
Per_PerformedActivityID <fi1> II
PerformedObservationID <fi1> II
Per_Per_PerformedActivityID <fi2> II
Per_PerformedObservationID <fi2> II
reportedDate Timestamp
targetAnatomicSiteLateralityCode CD
typeCode CD
uncertaintyCode CD
valueCodeModifiedText ST
valueNullFlavorReason ST
Identifier_1 <pi>
lationship_2
BRIDG HCT 1.0 Logical
PerformedObservationResultValue
codeValue <pi> CS <M>
identifier <fi> II
description ST
valueReal REAL
valueString ST
unitofMeasure ST
intervalStart REAL
intervalEnd REAL
Identifier_1 <pi>

We could have added all these attributes to the ‘PerformedObservationResult’ entity, but we had made the decision to create template tables for these complex datatypes. If the definition of ‘Any’ were to change in a later instantiation of BRIDG or BRIDG HCT, we wanted to manage the change first (in the conceptual model), and handle complex data types in the logical model.

We could have broken apart the ‘PerformedObservationResult’ entity into separate entities by data type. For example, we could have created entities of “PerformedObservationResultDate’, ‘PerformedObservationResultString’, ‘PerformedObservationResultNumber’, etc.
This would have required understanding the data type of the value in order to be able to populate the correct table. On the one hand this facilitates some analysis because one does not have to know the data type for a particular entry. However this is a relatively generic entity anyway, so one cannot simply add all numbers that exist in this table. One has to understand from the ‘DefinedObservation’ the actual context of the value given:

Table 3: Comparing two CDEs that have a number as their value

CDE	Question Text	Mapping Path	DefinedActivity
2874163	Height	Subject[Recipient] > PerformedObservation > PerformedClinicalResult.value WHERE PerformedClinicalResult > PerformedObservation > DefinedObservation.nameCode = "Measure height"	"Measure height"
2675071	What is the chronological number of the current hematopoietic	PerformedProcedure > AssessedResultRelationship > PerformedObservation > PerformedObservationResult.value WHERE PerformedObservationResult > PerformedObservation >	"Determine chronological number of Overall Hematopoietic Stem Cell Transplantation"

	stem cell transplantation?	DefinedObservation.nameCode = "Determine chronological number of Overall Hematopoietic Stem Cell Transplantation"	

We could have left the column as a BLOB (binary large object). Certainly a binary large object could contain any of the data types that we have been discussing. However most RDBMS system do not do a good job of parsing that information for the ease of analytics. Therefore we decided to strike some middle ground by having one ‘PerformedObservationResult’ table, and having a minimum set of attributes that we felt could store the data that we needed.

Physical Level:

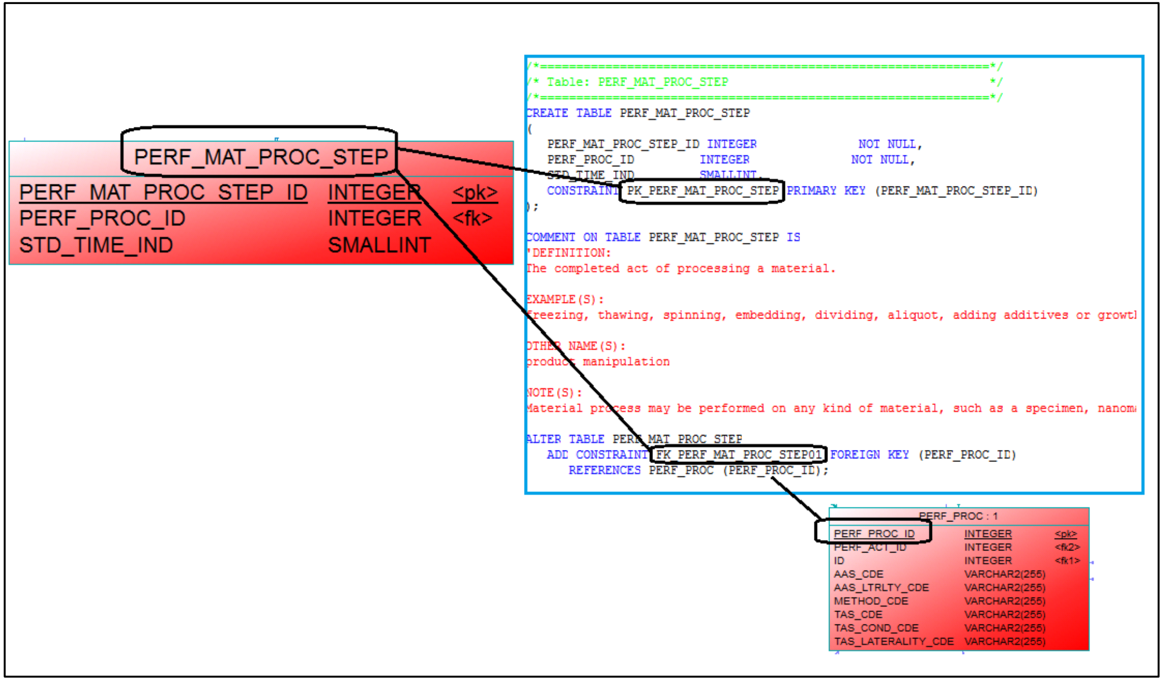
Naming Standards:
We decided to have a physical model conform to the widest variety of RDBMS available. This required that all objects at the physical level must conform to an ANSI standard. This meant that we would have no entities or attributes or any other object to being longer than 30 characters, and that we would not depend upon an RDBMS that could handle camel case. Therefore we accept the fact that these 30 character object name must also contain underscores in order to separate words. The list of abbreviations that we have used is included in the information set. we know that this is a working document and will change during the course of this modeling effort. This list of abbreviations is what the authors considered to be the roughest part of the physical model. We are aware of the pain of modifying column names and table names as abbreviations are incorporated into the list and have done their best to try to come up with a list that we feel is sufficient to contain the content as described and that can be extended.

The Current Naming standards were defined so that physical objects could be based on table and column names. This required a further shortening of the table and column names. The benefit is that one knows how database objects that are used to support a table or column will be named.

Table 4: Naming Conventions within BRIDG HCT 1.0

Object Type	**Maximum Length Allowed**	**Transformation Rule**
Table Name	25	
Column Name	25	
Primary Key Name	30	PK_{Table Name}
Alternate Key	30	AK_{Table Name}##
Index	30	IX_{table name}##
Primary Key Constraint	30	Same as Primary Key Name
Foreign Key Constraint	30	FK_{child table name}##

Figure 19: Example of applied naming conventions



PERF_MAT_PROC_STEP		
PERF_MAT_PROC_STEP_ID	INTEGER	<pk>
PERF_PROC_ID	INTEGER	<fk>
STD_TIME_IND	SMALLINT	

```
/*==============================================================*/
/* Table: PERF_MAT_PROC_STEP                                    */
/*==============================================================*/
CREATE TABLE PERF_MAT_PROC_STEP
(
   PERF_MAT_PROC_STEP_ID INTEGER              NOT NULL,
   PERF_PROC_ID         INTEGER              NOT NULL,
   STD_TIME_IND         SMALLINT,
   CONSTRAINT PK_PERF_MAT_PROC_STEP PRIMARY KEY (PERF_MAT_PROC_STEP_ID)
);

COMMENT ON TABLE PERF_MAT_PROC_STEP IS
'DEFINITION:
The completed act of processing a material.

EXAMPLE(S):
freezing, thawing, spinning, embedding, dividing, aliquot, adding additives or growt

OTHER NAME(S):
product manipulation

NOTE(S):
Material process may be performed on any kind of material, such as a specimen, nanom

ALTER TABLE PERF_MAT_PROC_STEP
   ADD CONSTRAINT FK_PERF_MAT_PROC_STEP01 FOREIGN KEY (PERF_PROC_ID)
      REFERENCES PERF_PROC (PERF_PROC_ID);
```

PERF_PROC : 1		
PERF_PROC_ID	INTEGER	<pk>
PERF_ACT_ID	INTEGER	<fk2>
ID	INTEGER	<fk1>
AAS_CDE	VARCHAR2(255)	
AAS_LTRLTY_CDE	VARCHAR2(255)	
METHOD_CDE	VARCHAR2(255)	
TAS_CDE	VARCHAR2(255)	
TAS_COND_CDE	VARCHAR2(255)	
TAS_LATERALITY_CDE	VARCHAR2(255)	

Table Row Identifiers

The decision on what constitutes a unique identifier for a role within a table is one where we need the most feedback from the community. We have we have declared the identity data type for a table as an integer. This is a classic decision for a table. However, the maintenance and integration of data between multiple organizations that wish to share data using this identity schema becomes very complex. It is possible to have one organization maintain all identifiers, and distribute the valid sets that can be used by multiple organizations, but it can be very difficult and very expensive to maintain. Since the BRIDG HCT 1.0 model is intended to be a community resource we do not consider the use of an integer as a data type to be sustainable under our constraints of resources within the CIBMTR and the community at large.

There are multiple options for identifiers for tables. An example of this is the HL7 OID.   Another example is to use a GUID (Global Unique Identifier). It is an ongoing topic of investigation and one where we need feedback from the community, to determine what is going to be most sustainable identifier for a role in the table within the BRIDG HCT Model.

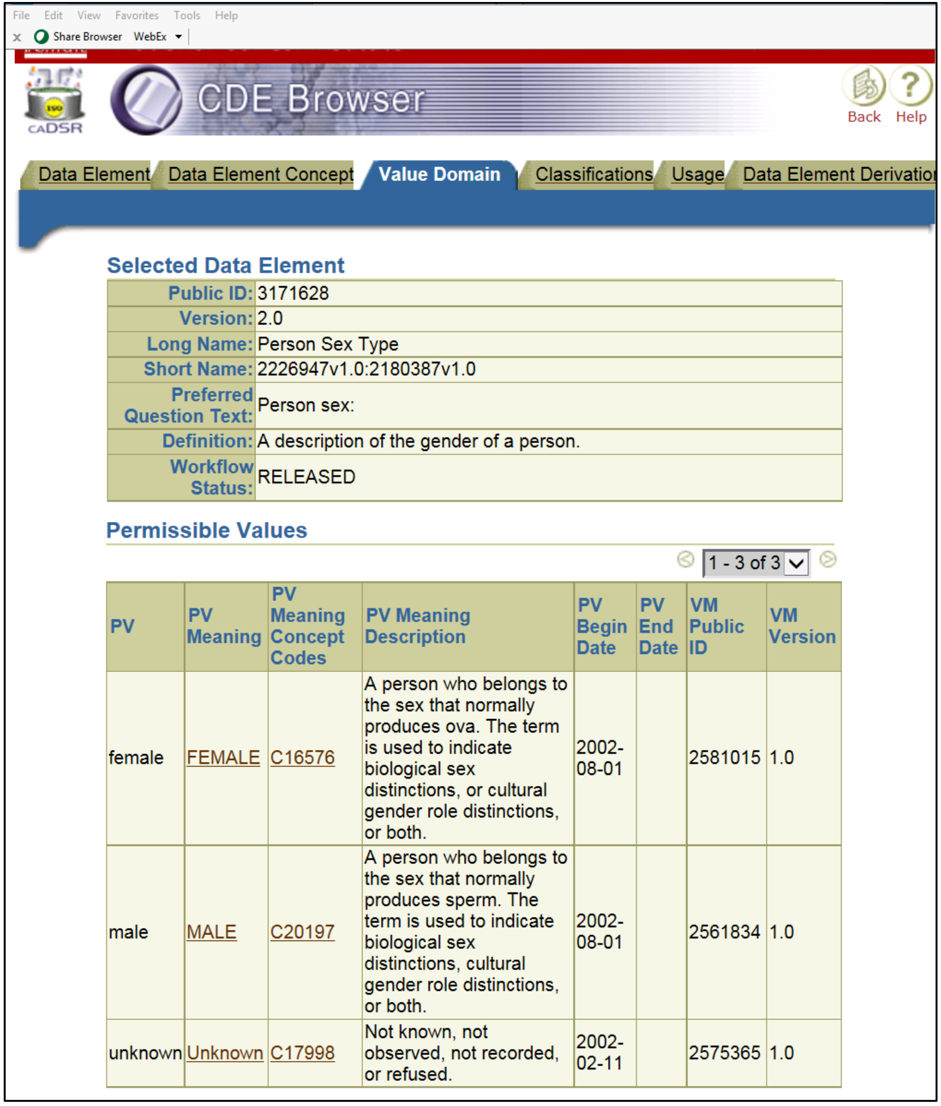
HL7 Implementation Guidance for Unique Object Identifiers:
http://www.hl7.org/documentcenter/private/standards_temp_2E1D25F2---1C23---BA17---0C74CBDB29844F8B/v3/V3_OIDS_R1_INFORM_2011NOV.pdf

Option Value Constraints.

There are many negative downstream effects of allowing data to be contained in a ‘typecode’ data type that is not validated against a set of permissible values. For example, if you have ‘typecode’ for person gender and you have no restriction on the set of option values, it is possible to add a value of ‘pineapple’ to a database, and then deal with the negative downstream consequences of fixing this data every place where the word ‘pineapple’ has been propagated. If a physical database does not allow the term ‘pineapple’ to be included as a valid gender for a person, this facilitates the ease of data transmission and data comprehension. The CDEs that are the source of all of the content within the BRIDG HCT model also have a strong definition of the permissible values for the CDEs. These restrictions need to be applied at the physical level for this BRIDG HCT 1.0 model.

The source of these ‘option values’ is the permissible-value list for each CDE within the caDSR:

Figure 20: Permissible Values for a CDE



Selected Data Element

Public ID:	3171628
Version:	2.0
Long Name:	Person Sex Type
Short Name:	2226947v1.0:2180387v1.0
Preferred Question Text:	Person sex:
Definition:	A description of the gender of a person.
Workflow Status:	RELEASED

Permissible Values

1 - 3 of 3

PV	PV Meaning	PV Meaning Concept Codes	PV Meaning Description	PV Begin Date	PV End Date	VM Public ID	VM Version
female	FEMALE	C16576	A person who belongs to the sex that normally produces ova. The term is used to indicate biological sex distinctions, or cultural gender role distinctions, or both.	2002-08-01		2581015	1.0
male	MALE	C20197	A person who belongs to the sex that normally produces sperm. The term is used to indicate biological sex distinctions, cultural gender role distinctions, or both.	2002-08-01		2561834	1.0
unknown	Unknown	C17998	Not known, not observed, not recorded, or refused.	2002-02-11		2575365	1.0

Within the domain analysis model, the conceptual model, and the logical model this data type has been denoted as ‘CD’. There are several hundred attributes within the model that have the data type of ‘CD’. We wished to apply a consistent mechanism of enforcing permissible values for this data type.

There were multiple ways to instantiate this data type at the physical level.

1. One method of instantiating this would be to create a separate table of id-value pairs. The identifier would be an arbitrary, immutable, and unique. This is typically instantiated with an integer value. This identifier would

be associated with a character string that would describe the value. Note that this is actually how these option values are stored within the caDSR. (See above: each permissible value for the gender types are associated with a unique identifier). It would be possible to extract this entire set of permissible values and have a unique set of constraints to be applied. However, it would also require the creation of hundreds of small tables within the BRIDG HCT 1.0 model that would need to be maintained.

2. Another method of enforcing valid values within a 'CD' data type is to create a database constraint with permissible values. This is the least attractive of the options. On the one hand, it uses the power of the RDBMS to enforce valid data within the database, on the other hand every time permissible value changes it requires a modification of the database itself. This can be difficult to maintain.
3. Another method of enforcing valid values within a 'CD' data type, it to use a database trigger. Upon the insertion of a role within a table, or upon the update of this particular 'CD' column within a table, it is possible to have the database validate the value being modified. In the BRIDG HCT 1.0 model, this is the mechanism that we will prototype. We will create one table that contains the context (the table and column name) and the valid set of option values. If a row exists in the table under the defined context, it is considered to be valid and the insertion or update will take place. If it does not exist, the trigger will execute an error. The trigger can be modified to log into error handling mechanism. Error-handling was not done for the BRIDG HCT 1.0 model purely due to time constraints.

The BRIDG HCT data modelers intend to create the same trigger for every attribute of the CD data type. This can be done through code, and therefore it is relatively easy to maintain. We are aware that there can be serious performance degradation due to the over-use of triggers and the interaction between database triggers. There are some institutions that do not allow triggers within their databases due to this reason. There are other ways of instantiating the same concept.

What is NOT in the physical model and should be:

Any physical model should pay attention to the following issues. They are not included in the BRIDG HCT 1.0 model, because they are very specific to the physical instantiation of any model, with regards to particular institutional internal policies on database creation. We will give you an example of how we apply them within the CIBMTR, but we do not include DDL (Data Definition Language) in the BRIDG HCT 1.0 model, because these are best left to the database administrators and database modelers at a given institution.

Physical storage:

The physical storage of the table and its attributes are very specific to the RDBMS at a particular institution. The choices one makes for a block size or the table space size for a vendor such as Oracle, are not the decisions that one would make for a Microsoft SQL database. Therefore none of those provisions are listed here. It is up to an individual database administrator to apply those policies for all objects within the model.

Indexing policies:

Indexing policies for tables are also very specific to a particular institution. We have included primary keys on the tables within the BRIDG HCT 1.0 model. Note that there are few alternate keys nor are there any other indexes for these tables. Some institutions have policies that indexes are not stored in the same physical container as the tables. For example, one can have an Oracle tablespace for data and a different Oracle tablespace for indexes. It is up to an individual database administrator to apply the specific organization's policies for all objects within the model.

Rights to Objects:

Note that there are no rights applied to any of the objects table. This means that only the owner of the particular database has rights to do anything to the objects within the table. This is also very specific to an organization's policies for rights to data. Some organizations define an application, and then apply specific rights to the tables needed for that application as necessary. Some will differentiate between specific roles within an organization such as administrators, analysts, and maintainers of data and apportion the ability to read and write the data in these tables accordingly. It is up to an individual database administrator to apply the specific organization's policies for all objects in the model.

Auditing Policies:

Note there is no audit trail for any of the objects within the model. Again, it is specific to how an organization wishes to track changes within the model. Depending on how this model is used it may be determined that no historical record is necessary. Some institutions may determine that they wish to keep a copy of all of the data within a table in an audit table within the model. Yet another organization may choose to keep track of the audit logs for a specific RDBMS vendor, and use those audit logs to create a copy of the data in another location. It is up to an individual database administrator to apply the specific organization's policies for all objects in the model.