	LRA Technical Model Developer's Manual Part 2: Model Construction			
	Programme	Informatics Data Standards	Document Record ID Key	
	Sub-Prog / Project	Logical Record Architecture for Health and Social Care	NPFIT-FNT-TO-DPM-1002.02	
	Prog. Director	D. Perry	Status	Approved
	Owner	S. Bentley	Version	1.0
	Author	D. McGuffin	Version Date	26/11/2009

Logical Record Architecture for Health and Social Care: LRA Technical Model Developer's Manual Part 2: Model Construction

Amendment History:

Version	Date	Amendment History
0.1	16/10/2009	First draft for comment
0.2	24/11/2009	Minor update following internal review.

Reviewers:

This document must be reviewed by the following:

Name	Signature	Title / Responsibility	Date	Version
Steve Bentley				
John Arnett				
Nicholas Oughtibridge				
Ed Cheetham				
Andrew Hinchley				
Matt Stephens				
Yongsheng Gao				

Approvals:

This document must be approved by the following:

Name	Signature	Title / Responsibility	Date	Version
Steve Bentley/ Laura Sato		Head of LRA	26/11/2009	1.0

Distribution:

<Author to say who the document will be distributed to>

Document Status:

This is a controlled document.

Whilst this document may be printed, the electronic version maintained in FileCM is the controlled copy. Any printed copies of the document are not controlled.

Related Documents:

These documents will provide additional information.

Ref no	Doc Reference Number	Title	Version
1	NPFIT-SHR-QMS-PRP-0015	Glossary of Terms Consolidated.doc	<enter latest>

Glossary of Terms:

List any new terms created in this document. Mail the NPO Quality Manager to have these included in the master glossary above [1].

Term	Acronym	Definition
------	---------	------------

--	--	--

Contents

1	About this Document	5
1.1	Purpose	5
1.2	Audience	5
1.3	Content	5
2	Prerequisites	6
2.1	Toolset	6
2.1.1	Installing SNOB Browser	6
2.2	Documentation Required	6
2.3	UML Profile for LRA Technical Modelling	7
2.4	LRA Technical Model Class Templates	7
2.5	LRA Constraint Types	8
3	Modelling Methodology	9
3.1	LRA Technical Model Artefacts	9
3.1.1	ELEMENT Domain Models	9
3.1.2	Participation Domain Models	13
3.1.3	ENTRY Domain Models	19
3.2	LRA Interface Model Artefacts	22
3.2.1	ELEMENT Constrained Domain Models	22
3.2.2	ENTRY Constrained Domain Models	27
3.2.3	Participation Constrained Domain Models	31
3.3	LRA SNOMED CT Expression Constraints	38
3.3.1	Semantic Expression Constraints	39
3.3.2	Literal Expression Constraints	39
4	Collaborative Working Approach	40
4.1	Collaborative Communication Log	40
A	Appendix A: LRA Constraint Types	43

1 About this Document

1.1 Purpose

The purpose of this document is to provide Technical Model authors with instructions on how to construct Technical Model artefacts for LRA content releases.

This document forms part 2 of a two part LRA Technical Model Developer's Manual. Part 1 of the manual provides detailed guidance on how to design Technical Model artefacts for LRA content releases.

1.2 Audience

The intended audience of this document is any person responsible for the authoring of LRA Technical or Interface Model Artefacts for Release 2.0.

1.3 Content

This document comprises the following sections:

- Prerequisites
- Modelling Methodology
 - LRA Technical Model Artefacts
 - LRA Interface Model Artefacts
 - LRA SNOMED CT Expression Constraints
- Collaborative Working Approach
- Appendices

2 Prerequisites

2.1 Toolset

The following tooling is required for the production of LRA Technical and Interface Model Artefacts:

- Sparx Enterprise Architect (Professional Edition) version 7.1 or later.
- A suitable terminology browser (e.g. CliniClue, SNOB Browser etc.) and relevant SNOMED CT release. Installation instructions for SNOB browser can be found below.

2.1.1 Installing SNOB Browser

SNOB browser provides an additional benefit of allowing the browsing of SNOMED CT subsets. Installation instructions are as follows:

1. Archives for the SNOB Browser and SNOMED CT release files (*SNOB 2.90.01 Install* and *SNOB 2.xx.xx Data - UK Release 20091001* respectively) are available at https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/app_data/SNOB.
2. Create a subdirectory on your machine (e.g. *C:/SNOB*) for the SNOB browser and SNOMED CT release files.
3. Unzip the file *SNOB 2.90.01 Install.zip* into this new directory.
4. Continue installation by following the instructions in the newly extracted *README SNOB 2.90.01 Install.txt* file.
5. Further instructions on the use of SNOM browser are available at <http://snob.eggbird.eu>.

2.2 Documentation Required

Access and familiarity with the latest version of the following documentation is required for the production of LRA Technical and Interface Model Artefacts:

- NPFIT-FNT-TO-DPM-0935 LRA Technical Model Infrastructure Specification Part 1: Care Components
(https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/infrastructure/doc/LRA_Care_Components_Specification.doc)
- NPFIT-FNT-TO-DPM-0922 LRA Technical Model Infrastructure Specification Part 2: Participations
(https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/infrastructure/doc/LRA_Participations_Specification.doc)
- NPFIT-FNT-TO-DPM-1001 LRA Technical Model Developer's Manual Part 1: Model Design
(https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/development/doc/LRA_Technical_Model_Design_Manual.doc)

- <TBD> Logical Record Architecture for Health and Social Care: Technical Model Representation, Constraint and Serialisation Specification (https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/infrastructure/doc/LRA_TM_RCS_Specification.doc)
- <TBD> LRA Technical Model Analysis and Design – Renal (https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/domains/renal/LRA_TM_Design-Renal.doc)
- BS EN 13606-1:2007 - Health informatics – Electronic health record communication - Part 1: Reference model ([https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/EN_13606/BS EN 13606.1.pdf](https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/EN_13606/BS_EN_13606.1.pdf))
- BS EN 13606-3:2008 - Health informatics – Electronic health record communication - Part 3: Reference archetypes and term lists ([https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/EN_13606/BS EN 13606.3.pdf](https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/EN_13606/BS_EN_13606.3.pdf))
- Object Constraint Language Version 2.0 Specification (https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/ref/03-10-14.pdf)

2.3 UML Profile for LRA Technical Modelling

The current UML profile for LRA Technical Modelling must be imported into EA as follows:

- Update / checkout UML profiles folder and contents from: https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/model/uml_profiles/ to SVN working copy;
- Select the Resources tab in EA (next to Project Browser), right click on UML Profiles folder in Resources tree and select Import Profile; and
- Navigate to the UML profiles folder in file browser, select the *EA_UML_Profile-LRA_Technical_Modelling* file and select open and then import.

After the import has completed the UML Profiles folder will be updated with a UML Profile for LRA Technical Modelling subfolder.

2.4 LRA Technical Model Class Templates

The current set of LRA Class Templates must be imported into EA as follows:

- Update / checkout UML patterns folder and contents from https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/model/uml_patterns/ to SVN working copy;
- Select the Resources tab in EA (next to Project Browser), right click on UML Patterns folder in resources tree and select Import UML Pattern; and
- Navigate to the UML patterns folder in file browser, multi-select all of the files and select open.

After the import has completed the UML Patterns folder will be updated with a LRA Class Templates subfolder.

2.5 LRA Constraint Types

The current set of LRA Constraint Types must be imported into EA as follows:

- Update / checkout Reference data folder and contents from https://svn.connectingforhealth.nhs.uk/svn/public/lra/TRUNK/prod_env/infrastructure/model/ref_data/ to SVN working copy;
- Select *Import Reference Data* from the *Tools* menu in EA; and
- Navigate to the Reference data folder in file browser, select the *EA_REF_General_Constraint_Types* file and select open and then import.

After the import has completed the LRA Constraint Types will be available in EA. More information on LRA Constraint Types is given in Appendix A.

3 Modelling Methodology

The approach to developing LRA Model Artefacts for LRA Release 2.0 requires collaborative working between members of the LRA knowledge modelling, technical modelling and terminology teams. Further details of this approach can be found in Section 4 below.

The production of LRA Model Artefacts for LRA Release 2.0 consists of the following stages.

3.1 LRA Technical Model Artefacts

3.1.1 ELEMENT Domain Models

An ELEMENT domain model is comprised of exactly one ELEMENT from the Care Components Reference Model along with any applicable COMPONENT_RELATIONSHIP_ELEMENTS. An applicable COMPONENT_RELATIONSHIP_ELEMENT is one where a SUBJECT association exists between it and the ELEMENT in question.

Each ELEMENT domain model exists in its own package (which has the same name as the ELEMENT class in Step 2a below) within the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Technical.ELEMENT Domain Models* package.

Prior to commencing construction of an ELEMENT Domain Model collaborative analysis of the data element content requirements by the technical modeller(s) and terminology expert(s) needs to be performed in order to agree the information model / terminology responsibilities.

An ELEMENT Domain Model is constructed as follows:

1. For each data element content requirement select the appropriate ELEMENT class from the LRA Class Templates resource folder. Guidance on the choice of ELEMENT classes is available in the *LRA Technical Model Developer's Manual Part 1: Model Design* document.
2. For each BOUND_DATA_ELEMENT or UNBOUND_DATA_ELEMENT class:
 - a. In the class Name field (of the class dialogue) enter a short name that describes the ELEMENT based on its data element content requirement. The name shall be formatted in camel case.
 - b. Set the Status field (of the class dialogue) to *Proposed*.
 - c. In the class Notes field (of the class dialogue) enter a description of the ELEMENT based on the provided data element content requirement. The description should take the following form:
 - An initial high level statement describing the purpose of the ELEMENT commencing with the text:
"This ELEMENT Domain Model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA."

An ELEMENT to represent...”; followed by

- Any other narrative text required to describe the purpose of this ELEMENT including a formal definition (if applicable).
- References to appropriate documentation shall be included and listed in the description.
- For example,

Class : StandingBodyHeightObservation

General Details Require Constraints Links Scenario Files

Name: StandingBodyHeightObservation

Stereotype: PROPERTY_OBSERVATION ☐ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A x^2 x_2

This ELEMENT Domain Model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA. An ELEMENT to represent the result of a standing body height measurement.

Standing body height is the distance from the bottom to the top of something standing upright; *especially*: the distance from the lowest to the highest point of an animal body especially of a human being in a natural standing position or from the lowest point to an arbitrarily chosen upper point^{1,3}. This excludes other measurements of length or distance^{1,3}.

¹: NPFIT-FNT-TO-DPM-0725.06 - NPFIT Message Implementation Manual 7.2.02
³: NPFIT-NCR-DES-0422.06 Representation of Commonly Used Concepts in the Care Record

- The ELEMENT class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The ELEMENT class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
- For each ELEMENT attribute:

- If the attribute is shown as readOnly then its value and properties are fixed by the model and must not be edited by the user.

Else:

- If the lower bound of the attribute multiplicity = 0 (e.g. 0..1, 0..* etc.):
 - Include the following attributes by setting the lower bound of the attribute multiplicity to 1 or more:
 - *<no attributes identified>*
 - Exclude the following attributes by setting the upper bound of the attribute multiplicity to 0. The display of any excluded attributes should be suppressed using the feature visibility (see EA help) of the class:
 - archetype_id
 - policy_ids
 - sensitivity
 - orig_parent_ref
 - The meaning attribute (MATERIAL_ENTITY_ELEMENT only) should be included or excluded in accordance with the data element content requirements.
 - The obs_time attribute (not MATERIAL_ENTITY_ELEMENT) should remain optional (i.e. attribute multiplicity of 0..1) as it is extremely unlikely that the modeller will be able to predict whether the obs_time or COMPOSITION.session_time will be applicable in an ELEMENT instance.
 - The value attribute (MATERIAL_ENTITY_ELEMENT only) should be included or excluded in accordance with the data element content requirements.
 - If required (and where permitted) to constrain the attribute type, enter a (non-abstract) specialisation of the data type into the Type field (in the attribute details dialogue); e.g. type: ANY might be constrained to type: ED.
 - In the attributes Notes field (of the attribute details dialogue) enter an applicable description of the attribute. Document references should be included where applicable.
 - Do not edit the Name, Alias, Scope, Stereotype, Initial, Derived, Static, Property or Const fields in the attribute details dialogue.
- f. For the meaning attribute (not UNBOUND_DATA_ELEMENTS):
- If required to further refine the SNOMED CT terminology binding, update the value of the code.semantic tagged value (via the Tagged Values tab in EA) with the relevant UUID. Further information on the specification of SNOMED CT Expression Constraints is available in Section 3.3 below.
- g. For the synthesised attribute:

- Specify the value.enum tagged value based on the data element content requirements.
- h. For the value attribute (PROPERTY_OBSERVATION_ELEMENTS, MATERIAL_ENTITY_ELEMENTS and UNBOUND_DATA_ELEMENTS only):
 - Specify the required tagged values for this attribute based on the data element content requirement, or the constrained attribute type chosen in previous step. For example, tagged values may be used to specify the units of a PQ or to specify a nullFlavor. Guidance on the use of tagged values is available in the *Technical Model Representation, Constraint and Serialisation Specification*.
- 3. For each COMPONENT_RELATIONSHIP_ELEMENT class (the UML Profile used in the LRA represents a COMPONENT_RELATIONSHIP_ELEMENT and subsequent contained subject and object LINK classes diagrammatically as a yellow diamond):
 - a. No action is required for either the class name or class notes field of this class.
 - b. Set the Status field (of the class dialogue) to *Proposed*.
 - c. Specify the meaning.code.enum tagged value of the COMPONENT_RELATIONSHIP_ELEMENT instance with a literal value from the LraComponentRelationshipElementMeaning value set. Guidance on the use of tagged values is available in the *Technical Model Representation, Constraint and Serialisation Specification*.
 - d. Set the object.follow_link.value.enum and subject.follow_link.value.enum tagged values to *true*.
 - e. Specify the value.code.enum tagged value as required.
 - f. Create the necessary associations between the COMPONENT_RELATIONSHIP_ELEMENT and its subject and object ELEMENTs. For each association:
 - The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the COMPONENT_RELATIONSHIP_ELEMENT and the target class shall be a BOUND_DATA_ELEMENT or UNBOUND_DATA_ELEMENT.
 - Each COMPONENT_RELATIONSHIP_ELEMENT instance shall only have one subject ELEMENT but may have one or more object ELEMENTs.
 - The stereotype field (in the association properties dialogue) of each association shall be specified as either SUBJECT or OBJECT appropriately.
 - The multiplicity field shall only be specified for the source role of each association (in the association properties dialogue).

3.1.2 Participation Domain Models

A Participation domain model is comprised of classes predominantly from the Participations Reference Model; there is also limited overlap and use of classes from the Care Components Reference Model.

Participations Domain Models describe the involvement of entities and roles within the health record. Examples of entities include person, device and organisation, whereas examples of roles include patient, care professional and hospital departments. These entities and roles may participate in the health record in various ways (e.g. as authors, performers, subject of the care record etc.).

Within Participations Domain Models, classes from the Participations Reference Model serve to describe the details of any entities and roles. Classes from the Care Components Reference Model act as containers for choices of Participations Reference Model classes (e.g. a FUNCTIONAL_ROLE class may serve as a container for a choice of combinations of Participations Reference Model class; each combination representing the details of an information provider).

Each Participation domain model class exists in its own package in the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Technical.Participation Domain Models* package. The only exception to this rule is for classes that have a stereotype of *CR_RoleRelationship* which exist in the same package as the associated *CR_Role* class with the *source* association role name.

A Participation Domain Model is constructed as follows.

1. For each class required from the Participations Reference Model:
 - a. Select the relevant Participations Reference Model class from the LRA Class Templates folder. Participations Reference Model classes are identifiable by the prefix "CR_" with the exception of the *CR_Participation* class which belongs to the Care Components Reference Model.
 - b. In the class Name field (of the class dialogue) enter a short name that describes the Participation class based on its data element content requirements. The name shall be formatted in camel case.
 - c. Set the Status field (of the class dialogue) to *Proposed*.
 - d. In the class Notes field (of the class dialogue) enter a description of the Participation class based on the provided data element content requirement.
 - e. If the Participations class has a "*CR_Role...*" stereotype then that class should be made "composite" (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The Participations class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
 - f. For each Participations class attribute:
 - Set the multiplicity of the attribute according to the data element content requirements and the overall requirements of other associated

Participations classes. If possible the attribute should either be explicitly included (lower bound of attribute to 1 or more) or excluded (upper bound of attribute to 0) in the model.

- If required (and where permitted) to constrain the attribute type, enter a (non-abstract) specialisation of the data type into the Type field (in the attribute details dialogue); e.g. type: ANY might be constrained to type: ED.
 - In the attributes Notes field (of the attribute details dialogue) enter an applicable description of the attribute.
 - Do not edit the Name, Alias, Scope, Stereotype, Initial, Derived, Static, Property or Const fields in the attribute details dialogue.
 - If constraints are required to describe the behaviour of an attribute (e.g. conditional presence based on the inclusion of an attribute in another class), they should be captured at the level of the appropriate class (i.e. within the constraints tab of the Class dialogue) and not directly against the attribute. See next step for further details.
- g. Create any OCL constraints required to describe the use of this class or its attributes in the Constraints tab of the class dialogue. For each constraint:
- Select an appropriate constraint type in the Type field. Further information on LRA Constraint Types can be found in Appendix A.
 - Set the Status field to *Proposed*.
 - In the Constraint field enter an applicable name for the constraint.
 - Enter a descriptive comment and the constraint expression into the main field.
- h. Create any required associations between this Participations class and any other Participations classes in order to fulfil the data element content requirements. All associations created should be compliant with the Participations Reference Model. For each association:
- Include a relevant description of the association (in the Notes field of the association properties dialogue).
 - Within the Source Role tab of the association properties dialogue specify a role name (when appropriate) and multiplicity in accordance with the data element content requirements.
 - Within the Target Role tab of the association properties dialogue specify a role name (when appropriate) and multiplicity in accordance with the data element content requirements.
- i. Complete the child diagram of each composite class:
- Select all classes required for the domain model by dragging from the Project browser.
 - For non-composite classes - suppress the visibility of any excluded attributes using the feature visibility (see EA) of the class.

- For composite classes (that are not the focal class / entry point to this child diagram) – suppress the visibility of all attributes and constraints using the feature visibility (see EA) of the class.
- Suppress the visibility of any non-applicable associations using the visibility (see EA) of the association.

2. For each class required from the Care Components Reference Model:

- a. Select a FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE class from the LRA Class Templates folder.
 - In the class Name field (of the class dialogue) enter a short name that describes the class based on its data element content requirements. The name shall be formatted in camel case.
 - Select the Abstract field (of the class dialogue).
 - Set the Status field (of the class dialogue) to *Proposed*.
 - In the class Notes field (of the class dialogue) enter a description of the class based on the provided data element content requirement, commencing:
 - “Represents a choice of <Specify class> classes used to describe the participation of a CR_Role class in an ENTRY. The CR_Role class shall participate...”
 - For example,

Class : OtherParticipationsChoice

General | Details | Require | Constraints | Links | Scenario | Files

Name: OtherParticipationsChoice

Stereotype: FUNCTIONAL_ROLE ☐ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A x^2 x_2

Represents a choice of FUNCTIONAL_ROLE classes used to describe the participation of a CR_Role class in an ENTRY. The CR_Role class shall participate as follows:

- as a non-specific 'other participation'. This option is applicable when it is not relevant to explicitly state the participation type or when the source system is unable to support further specification of the participation type.
- as any other explicitly stated participation type from the relevant vocabulary. This option shall not be used to describe a participation type of Information Provider (a.k.a. Informant).

- The class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
- b. For each FUNCTIONAL_ROLE_CHOICE class select a FUNCTIONAL_ROLE and/or CR_Participation class from the LRA Class Templates folder.
- Else:

For each RELATED_PARTY_CHOICE class select a RELATED_PARTY class from the LRA Class Templates folder.

- In the class Name field (of the class dialogue) enter a short name that describes the class based on its data element content requirements. The name shall be formatted in camel case.
- Select the Abstract field (of the class dialogue).
- Set the Status field (of the class dialogue) to *Proposed*.

- In the class Notes field (of the class dialogue) enter a description of the class based on the provided data element content requirement.
- For each class attribute:
 - If lower bound of attribute multiplicity = 0 (e.g. 0..1, 0..* etc.) either set the lower bound to 1 or more (to include the attribute in the model) or set the upper bound to 0 (to exclude the attribute from the model). The display of any excluded attributes should be suppressed using the feature visibility (see EA help) of the class.
 - If required (and where permitted) to constrain the attribute type, enter a (non-abstract) specialisation of the data type into the Type field (in the attribute details dialogue); e.g. type: ANY might be constrained to type: ED.
 - In the attributes Notes field (of the attribute details dialogue) enter an applicable description of the attribute.
 - Do not edit the Name, Alias, Scope, Stereotype, Initial, Derived, Static, Property or Const fields in the attribute details dialogue.
- c. Create a realization association between the FUNCTIONAL_ROLE_CHOICE and any FUNCTIONAL_ROLE or CR_Participation classes, or between the RELATED_PARTY_CHOICE and any RELATED_PARTY classes. For each realization association:
 - The source class shall be a FUNCTIONAL_ROLE, CR_Participation or RELATED_PARTY class.
 - The target class shall be the FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE class.
- d. For each FUNCTIONAL_ROLE, CR_Participation or RELATED_PARTY class create a CR_Role_Choice class from the LRA Class Templates folder.
 - In the class Name field (of the class dialogue) enter a short name that describes the class based on its data element content requirements. The name shall be formatted in camel case.
 - Select the Abstract field (of the class dialogue).
 - Set the Status field (of the class dialogue) to *Proposed*.
 - In the class Notes field (of the class dialogue) enter a description of the class based on the provided data element content requirement, commencing:
 - “Represents a choice of CR_Role classes when a CR_Role is being used to describe the details of an entity that participates...”
 - For example,

Class : OtherRoleChoice1

General Details Require Constraints Links Scenario Files

Name: OtherRoleChoice1

Stereotype: CR_Role_Choice ☒ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Advanced

Notes:

B I U A x^2 x_2

Represents a choice of CR_Role classes when a CR_Role is being used to describe the details of an entity that participates in a non-specific 'other participation' manner.

OK Cancel Apply Help

- The class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
- Within the child diagram of the CR_Role_Choice class:
 - Select any CR_Role specialization classes from the Project Browser in accordance with the data element content requirements.
 - Create a realization association between the CR_Role_Choice class and each CR_Role specialization class. For each realization association:
 - The source class shall be a CR_Role specialization class.
 - The target class shall be the CR_Role_Choice class.
 - Suppress the visibility of all attributes and constraints of each CR_Role specialization class using the feature visibility (see EA) of the class.

- e. Create an association between the FUNCTIONAL_ROLE, CR_Participation or RELATED_ROLE class and any CR_Role_Choice classes. For each association:
 - The source class shall be the FUNCTIONAL_ROLE, CR_Participation or RELATED_ROLE class.
 - The target class shall be a CR_Role_Choice class.
 - Set the Direction field (in the dependency properties dialogue) to *Source -> Destination*.

3.1.3 ENTRY Domain Models

An ENTRY domain model is comprised of exactly one ENTRY from the Care Components Reference Model.

Each ENTRY contains (as ELEMENTs) the information acquired and recorded for a single observation or observation-set (battery or time series), a single clinical statement such as a portion of the patient's history or an inference or assertion, or a single action that is intended or has actually been performed. Each Entry Domain Model contains one or more ELEMENT Domain Models.

Each ENTRY domain model exists in its own package in the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Technical.ENTRY Domain Models* package.

An ENTRY Domain Model is constructed as follows:

1. Select an ENTRY class from the LRA Class Templates resource folder.
2. In the class Name field (of the class dialogue) enter a short name that describes the ENTRY based on its data element content requirement. The name shall be formatted in camel case.
3. Set the Status field (of the class dialogue) to *Proposed*.
4. In the class Notes field (of the class dialogue) enter a description of the ENTRY based on the data element content requirement. The description should take the following form:
 - a. An initial high level statement describing the purpose of the ENTRY commencing with the text:

“This ENTRY domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

An ENTRY containing a collection of data items required to represent...”; followed by
 - b. A description of the types of ELEMENTs that are contained within this ENTRY commencing with the text “Comprises...”.
 - c. Any other narrative text required to describe the purpose and/or use-cases fulfilled by this ENTRY (Note that as ENTRYs are semantically neutral in the

LRA, formal definitions should not be included in the description of an ENTRY).

- d. A bulleted list of the data element content requirements used to create this ENTRY.
- e. References to appropriate documentation shall be included and listed in the description.
- f. For example,

Class : Temperature

General Details Require Constraints Links Scenario Files

Name: Temperature

Stereotype: ENTRY

Author:

Scope: Public

Alias:

Persistence:

Phase: 1.0 Version: 1.0

Status: Proposed

Complexity: Easy

Language: Java

Keywords:

☐ Abstract

Advanced

Notes:

B I U A | x^2 x_2

This ENTRY domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

An ENTRY containing a collection of data items required to represent the body temperature of the subject of the record.

Comprises a single coded Body Temperature Observation and optionally may contain:

- a textual description of the event
- a coded representation of the body temperature measurement procedure
- any coded representations of findings or observations that are relevant to this body temperature observation.

Body temperature shall be measured in degrees Celsius with accuracy to the nearest 0.1 degree³.

Incorporates requirements from the following sources:

- NPFIT MIM 7.2.02 Temperature template (COCD_TP146016UK04)
- Section 6 of NPFIT-NCR-DES-0422.06 Representation of Commonly Used Concepts in the Care Record
- Body Temperature Detailed Clinical Model (DCM) v0.7

³: NPFIT-NCR-DES-0422.06 Representation of Commonly Used Concepts in the Care Record

OK Cancel Apply Help

5. The ENTRY class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The ENTRY class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
6. For each ENTRY attribute:

- a. If the attribute is shown as readOnly then its value and properties are fixed by the model and must not be edited by the user.

Else:

- b. If lower bound of attribute multiplicity = 0 (e.g. 0..1, 0..* etc.) either set the lower bound to 1 or more (to include the attribute in the model) or set the upper bound to 0 (to exclude the attribute from the model). The display of any excluded attributes should be suppressed using the feature visibility (see EA help) of the class.
 - c. No further changes are permitted (or required) for ENTRY attributes.
7. Select all ELEMENT Domain Models (including COMPONENT_RELATIONSHIP_ELEMENTS) to be contained by this ENTRY (by dragging from the Project Browser).
 8. Create an association between the ENTRY instance and each ELEMENT instance (excluding COMPONENT_RELATIONSHIP_ELEMENTS). For each association:
 - a. The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the ENTRY instance and the target class an ELEMENT instance.
 - b. Set the value of the Direction field (in the association properties dialogue) to 'Source -> Destination'.
 - c. Populate the Notes field (in the association properties dialogue) with the following text "This association contains the data structure and values of the ENTRY. The items collection is constrained for use by the LRA to contain member instances of type ELEMENT only."
 - d. Select the Source Role tab and set the value of the Aggregation field to 'composite'. Note, it is intentional that a containment association is created in this manner as opposed to selecting a containment association directly from within the Class toolbox.
 - e. Select the Target Role tab of the association properties dialogue and:
 - Populate the Text Role field with the text "items".
 - Select an appropriate value from the Multiplicity field.
 - Select 'Private' from the Access field
 9. Select all Participation Domain Model classes to be included in this ENTRY (by dragging from the Project Browser). The selection of Participation Domain Models shall be limited to the following composite classes:
 - InformationProviderChoice
 - OtherParticipationsChoice
 - RelatedPartyChoice
 10. Create an association between the ENTRY instance and each Participation class instance. Note, that although there is a containment association between ENTRYs and FUNCTIONAL_ROLES / RELATED_PARTYs in the Care

Components model, the UML Profile used in the LRA represents these relationships as regular associations. For each association:

- a. The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the ENTRY instance and the target class a Participation Domain Model instance.
- b. Set the value of the Direction field (in the association properties dialogue) to 'Source -> Destination'.
- c. Select the Target Role tab (in the association properties dialogue) and populate the Multiplicity field with an appropriate value whilst remaining compliant with the Care Components model.

3.2 LRA Interface Model Artefacts

LRA Interface Model Artefacts are refinements of LRA Technical Model Interface Artefacts to reflect the requirements specified in the knowledge space for a particular domain.

Within EA the refinement is achieved by making each Interface class a specialization of the relevant Technical class. From a purely presentational perspective please ensure that for each Interface Model Artefacts diagram that the 'show parent' option is selected. This is achieved by selecting the *Show Additional Parents* field on the Diagram tab within the diagram properties dialogue.

3.2.1 ELEMENT Constrained Domain Models

An ELEMENT Constrained Domain Model is a refinement of an ELEMENT Domain Model to reflect the requirements specified in the knowledge space for a particular domain.

ELEMENT constrained domain models exist in the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Interface.<ENTRY>* package (where <ENTRY> specifies the ENTRY constrained domain model in which this model belongs).

Prior to commencing construction of an ELEMENT Constrained Domain Model collaborative analysis of the data element content requirements by the technical modeller(s) and terminology expert(s) needs to be performed in order to agree the information model / terminology responsibilities.

An ELEMENT Constrained Domain Model is constructed as follows; for each BOUND_DATA_ELEMENT or UNBOUND_DATA_ELEMENT:

1. Identify the Candidate Data Definition to be realized by an ELEMENT Constrained Domain Model from the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Knowledge.LRA Analysis.Data Element Definitions Palette.Candidate Data Elements.<Domain>* package (where <Domain> specifies the knowledge domain in which this model exists).
2. Identify the appropriate ELEMENT Domain Model to be refined from the *Logical Record Architecture for Health and Social Care.LRA Model*

Artefacts. Technical.ELEMENT Domain Models package. If no suitable ELEMENT Domain Model exists one should be created as per the instructions in Section 3.1.1 prior to continuing these steps. Note, the ELEMENT Constrained Domain Model being created in subsequent steps must be a valid refinement of the ELEMENT Domain Model identified at this stage.

3. Select the appropriate Constrained ELEMENT class from the LRA Class Templates resource folder:
 - a. Make the Constrained ELEMENT class a specialization of the ELEMENT Domain Model identified in Step 2 above.
 - b. In the class Name field (of the class dialogue) enter a short name that describes the Constrained ELEMENT based on its data element content requirement. The name shall be formatted in camel case.
 - c. Set the Status field (of the class dialogue) to *Proposed*.
 - d. In the class Notes field (of the class dialogue) enter a description of the Constrained ELEMENT based on the data element content requirement. The description should take the following form:
 - An initial high level statement describing the purpose of the ELEMENT commencing with the text:

“This Constrained ELEMENT domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

A constrained ELEMENT to represent...”; followed by
 - Any other narrative text required to describe the purpose of this ELEMENT including a formal definition (if applicable).
 - References to appropriate documentation shall be included and listed in the description.
 - e. For example,

Class : WeightObservation-Renal

General Details Require Constraints Links Scenario Files

Name: WeightObservation-Renal

Stereotype: PROPERTY_OBSERVAT ☐ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A x^2 x_2

This Constrained ELEMENT domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.
A constrained ELEMENT for the Renal Domain to represent the result of a body weight measurement.

4. The Constrained ELEMENT class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The Constrained ELEMENT class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
5. Constrain the class attributes for domain use as follows:
 - Annotate any attributes that require refined descriptions by appending the relevant text into the class description created in Step 3d above (i.e. within the class Notes field of the class dialogue). For example:

Class : HaemodialysisCatheterInsertion

General Details Require Constraints Links Scenario Files

Name: HaemodialysisCatheterInsertion

Stereotype: GENERAL_ACTIVIT ☐ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A $\frac{1}{2}$ $\frac{1}{3}$ x^2 x_2

This Constrained ELEMENT domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

A constrained ELEMENT to represent the procedure of inserting a catheter for haemodialysis.

The meaning attribute shall be used to represent the procedure and access site (including laterality) of inserting the haemodialysis catheter.

The obs_time attribute shall be used to represent the date and time of inserting the haemodialysis catheter if it differs from the session_time.

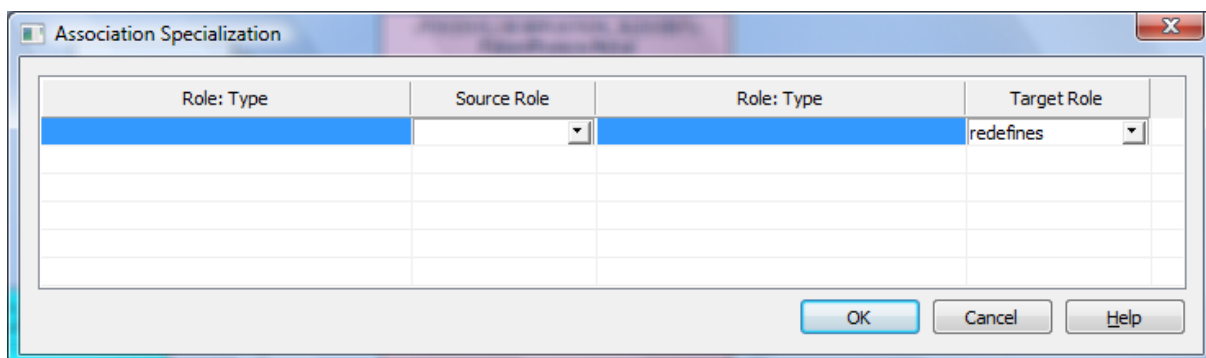
- To constrain the attribute type create an OCL constraint (using an *oclIsTypeOf(<ISO 21090 data type>)* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.
 - Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- To constrain the attribute multiplicity create an OCL constraint (using an *oclIsDefined()* or *oclIsUndefined()* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.
 - Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- To constrain a tagged value create a new tagged value for the class and:

- Prefix the tag name with the attribute name (e.g. *unit.enum* becomes *value.unit.enum*).
- Specify the tag value as appropriate.

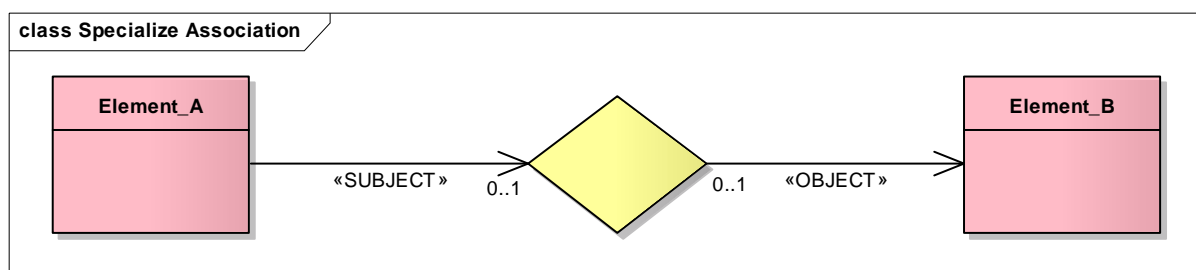
For each COMPONENT_RELATIONSHIP_ELEMENT required:

1. Identify the appropriate COMPONENT_RELATIONSHIP_ELEMENT class to be refined from the ELEMENT domain model identified above:
2. Select a COMPONENT_RELATIONSHIP_ELEMENT from the LRA Class Templates resource folder. Note, there is intentionally no Constrained COMPONENT_RELATIONSHIP_ELEMENT available in the LRA Class Templates resource folder.
 - a. No action is required for either the class name or class notes field of this class.
 - b. Set the Status field (of the class dialogue) to *Proposed*.
 - c. Specify the meaning.code.enum tagged value of this Constrained COMPONENT_RELATIONSHIP_ELEMENT instance with a literal value.
 - d. Specify the object.follow_link.value.enum, subject.follow_link.value.enum and value.code.enum tagged values.
 - e. Make the Constrained COMPONENT_RELATIONSHIP_ELEMENT class a specialization of the COMPONENT_RELATIONSHIP_ELEMENT class identified in Step 1 above
 - f. Create the necessary associations between this Constrained COMPONENT_RELATIONSHIP_ELEMENT and its subject and object Constrained ELEMENTs. For each association:
 - The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the Constrained COMPONENT_RELATIONSHIP_ELEMENT and the target class shall be a Constrained BOUND_DATA_ELEMENT or Constrained UNBOUND_DATA_ELEMENT.
 - Each Constrained COMPONENT_RELATIONSHIP_ELEMENT instance shall only have one subject Constrained ELEMENT but may have one or more object Constrained ELEMENTs.
 - The direction field shall be specified as *Bi-Directional*, *Source -> Destination* or *Destination -> Source*.
 - The stereotype field (in the association properties dialogue) of each association shall be specified as either SUBJECT or OBJECT appropriately.
 - The multiplicity field shall only be specified for the source role of each association (in the association properties dialogue).
 - The direction of the association shall be further specialized by right clicking on the association and selecting Advanced | Specialize Associations and

then selecting *redefines* for the relevant association role within the Association Specialization dialogue (see screenshot below).



For example, if the direction of the associations in the diagram below were to be specialized so that navigation was only possible from *Element_A* to *Element_B* (via the COMPONENT_RELATIONSHIP_ELEMENT) then the source association role would be redefined for the association stereotyped *SUBJECT* and the target association role would be redefined for the association stereotyped *OBJECT*.



3.2.2 ENTRY Constrained Domain Models

An ENTRY Constrained Domain Model is a refinement of an ENTRY Domain Model to reflect the requirements specified in the knowledge space for a particular domain.

ENTRY constrained domain models exist in their own subpackage within the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Interface* package.

An ENTRY Constrained Domain Model is constructed as follows:

1. #TODO Identify relevant knowledge artefact
2. Identify the appropriate ENTRY Domain Model to be refined from the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Technical.ENTRY Domain Models* package. If no suitable ENTRY Domain Model exists one should be created as per the instructions in Section 3.1.3 prior to continuing these steps. Note, the ENTRY Constrained Domain Model being created in subsequent steps must be a valid refinement of the ENTRY Domain Model identified at this stage.
3. Select a CONSTRAINED_ENTRY from the LRA Class Templates resource folder:
 - a. Make the CONSTRAINED_ENTRY class a specialization of the ENTRY Domain Model identified in Step 2 above.

- b. In the class Name field (of the class dialogue) enter a short name that describes the CONSTRAINED_ENTRY based on its data element content requirement. The name shall be formatted in camel case.
- c. Set the Status field (of the class dialogue) to *Proposed*.
- d. In the class Notes field (of the class dialogue) enter a description of the CONSTRAINED_ENTRY based on the data element content requirement. The description should take the following form:
 - An initial high level statement describing the purpose of the CONSTRAINED_ENTRY commencing with the text:

“This Constrained ENTRY domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

A constrained ENTRY containing a collection of data items required to represent...”; followed by
 - A description of the types of Constrained ELEMENTs that are contained within this Constrained ENTRY commencing with the text “Comprises...”.
 - Any other narrative text required to describe the purpose and/or use-cases fulfilled by this Constrained ENTRY (Note that as ENTRYs are semantically neutral in the LRA, formal definitions should not be included in the description of an ENTRY).
 - A bulleted list of the data element content requirements used to create this ENTRY.
 - References to appropriate documentation shall be included and listed in the description.
 - For example,

Class : Temperature-Renal

General Details Require Constraints Links Scenario Files

Name: Temperature-Renal

Stereotype: ENTRY ☐ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A $\frac{1}{2}$ $\frac{1}{3}$ x^2 x_2

This Constrained ENTRY domain model is included for illustrative purposes. The structure and terminology binding will be subject to change in forthcoming releases of the LRA.

A constrained ENTRY for the Renal Domain containing a collection of data items required to represent the body temperature of the subject of the record.

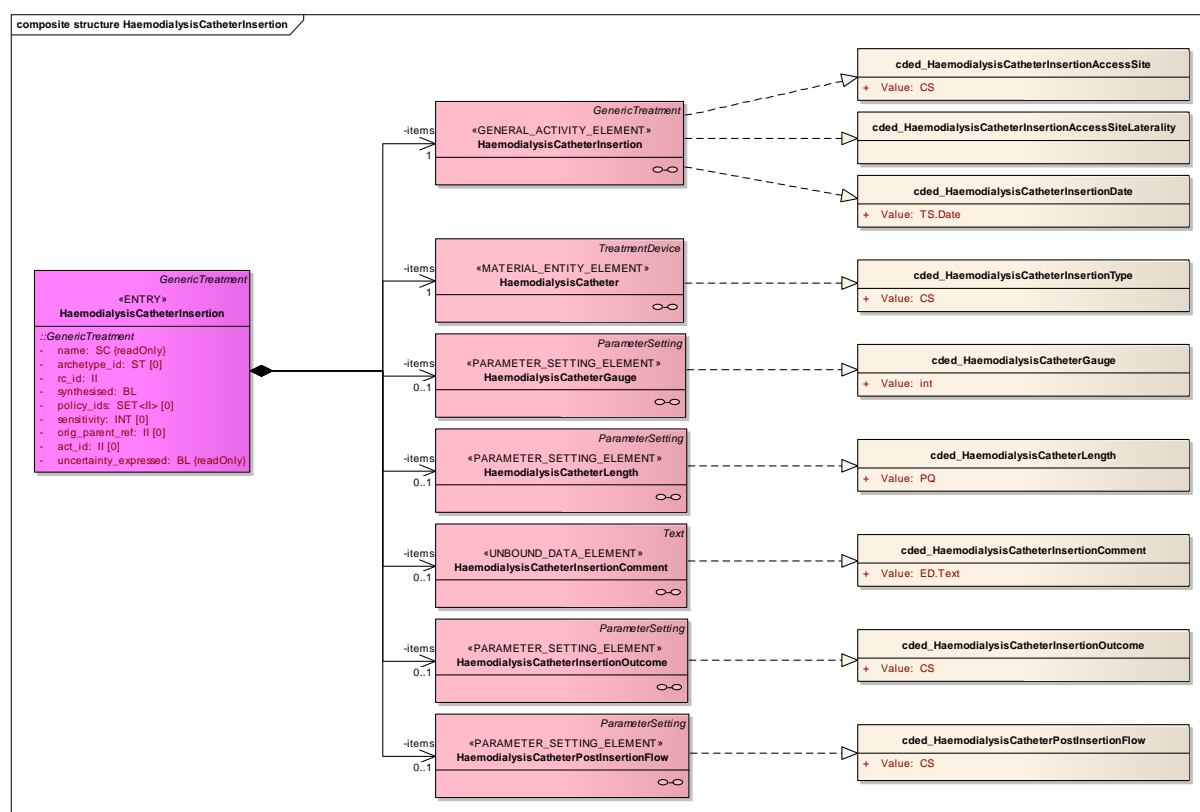
Comprises a single coded Body Temperature Observation.

4. Constrain the class attributes for domain use as follows:

- Annotate any attributes that require refined descriptions by appending the relevant text into the class description (i.e. within the class Notes field of the class dialogue).
- To constrain the attribute type create an OCL constraint (using an *oclIsTypeOf(<ISO 21090 data type>)* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.
 - Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- To constrain the attribute multiplicity create an OCL constraint (using an *oclIsDefined()* or *oclIsUndefined()* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.

- Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
 - To constrain a tagged value create a new tagged value for the class and:
 - Prefix the tag name with the attribute name (e.g. *unit.enum* becomes *value.unit.enum*).
 - Specify the tag value as appropriate.
5. The Constrained ENTRY class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The Constrained ENTRY class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
 6. Select all Constrained ELEMENT Domain Models (excluding COMPONENT_RELATIONSHIP_ELEMENTS) to be contained by this Constrained ENTRY (by dragging from the Project Browser).
 7. Create an association between the Constrained ENTRY instance and each Constrained ELEMENT instance. For each association:
 - a. The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the ENTRY instance and the target class an ELEMENT instance.
 - b. Set the value of the Direction field (in the association properties dialogue) to ‘Source -> Destination’.
 - c. Populate the Notes field (in the association properties dialogue) with the following text “This association contains the data structure and values of the ENTRY. The items collection is constrained for use by the LRA to contain member instances of type ELEMENT only.”
 - d. Select the Source Role tab and set the value of the Aggregation field to ‘composite’. Note, it is intentional that a containment association is created in this manner as opposed to selecting a containment association directly from within the Class toolbox.
 - e. Select the Target Role tab of the association properties dialogue and:
 - Populate the Text Role field with the text “items”.
 - Select an appropriate value from the Multiplicity field.
 - Select ‘Private’ from the Access field.
 8. Select all relevant Candidate Data Definitions (i.e. those identified in Step 1 in Section 3.2.1 above) that are realized by ELEMENTs in this ENTRY by dragging from the Project Browser. For each Candidate Data Definition create a realizes

association between the relevant ELEMENT and Candidate Data Definitions. For example:



9. Add any required Participations constrained domain models to this ENTRY constrained domain model as described in Section 3.2.3 below.

3.2.3 Participation Constrained Domain Models

A Participation Constrained Domain Model is a refinement of a Participation Domain Model to reflect the requirements specified in the knowledge space for a particular domain.

Participations constrained domain models exist in the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Interface.<ENTRY>* package (where **<ENTRY>** specifies the ENTRY constrained domain model in which this model belongs).

A Participations Constrained Domain Model is constructed as follows:

1. Identify the Candidate Data Definition to be instantiated as a Participation Constrained Domain Model from the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Knowledge.LRA Analysis.Data Element Definitions Palette.Candidate Data Elements.<Domain>* package (where **<Domain>** specifies the knowledge domain in which this model exists).
2. Identify the appropriate Participation Domain Model to be refined from the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Technical.Participation Domain Models* package. If no suitable Participation Domain Model exists one should be created as per the instructions

in Section 3.1.2 prior to continuing these steps. Note, the Participation Constrained Domain Model being created in subsequent steps must be a valid refinement of the Participation Domain Model identified at this stage.

3. Identify the ENTRY constrained domain model in which this Participation constrained domain model will exist.
4. Within the child diagram of the ENTRY constrained domain model identified in Step 3 above:
 - Select either a FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE class from the LRA Class Templates resource folder as indicated by the data element content requirements.
 - Make the selected class a specialization of the relevant Participation domain model class identified in Step 2 above.
 - In the class Name field (of the class dialogue) enter a short name that describes the class based on its data element content requirements. The name shall be formatted in camel case.
 - Select the Abstract field (of the class dialogue).
 - Set the Status field (of the class dialogue) to *Proposed*.
 - In the class Notes field (of the class dialogue) enter a description of the class based on the provided data element content requirement, commencing:
 - “Represents a choice of <Specify class> classes used to describe the participation of a CR_Role class in an ENTRY. The CR_Role class shall participate...”
 - For example,

Class : OtherParticipationsChoice

General Details Require Constraints Links Scenario Files

Name: OtherParticipationsChoice

Stereotype: FUNCTIONAL_ROLE ☒ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0 Advanced

Notes:

B I U A x^2 x_2

Represents a choice of FUNCTIONAL_ROLE classes used to describe the participation of a CR_Role class in an ENTRY. The CR_Role class shall participate as follows:

- as a non-specific 'other participation'. This option is applicable when it is not relevant to explicitly state the participation type or when the source system is unable to support further specification of the participation type.
- as any other explicitly stated participation type from the relevant vocabulary. This option shall not be used to describe a participation type of Information Provider (a.k.a. Informant).

OK Cancel Apply Help

The class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).

- Create an association between the ENTRY constrained domain model class and the FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE class. For each association:
 - The source class (in terms of the context of the relationship between the 2 classes in EA) shall be the ENTRY instance and the target class the FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE.
 - Set the value of the Direction field (in the association properties dialogue) to ‘Source -> Destination’.
 - Select the Target Role tab (in the association properties dialogue) and populate the Multiplicity field with the required value.
5. Within the child diagram of the FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE composite class identified in Step 4 above:

- For each FUNCTIONAL_ROLE_CHOICE class select a CONSTRAINED_FUNCTIONAL_ROLE and/or Constrained_CR_Participation class from the LRA Class Templates folder.

Else:

For each RELATED_PARTY_CHOICE class select a CONSTRAINED_RELATED_PARTY class from the LRA Class Templates folder.

- Make the selected class a specialization of the relevant Participation domain model class identified in Step 2 above.
- In the class Name field (of the class dialogue) enter a short name that describes the Participation class based on its data element content requirements. The name shall be formatted in camel case.
- Set the Status field (of the class dialogue) to *Proposed*.
- If the class selected was a CONSTRAINED_FUNCTIONAL_ROLE enter the following text in the class Notes field (of the class dialogue): "This class is used to document the participation of a person, device or software component in some activity recorded in the EHR."

Else:

- If the class selected was a Constrained_CR_Participation enter the following text in the class Notes field (of the class dialogue): "A class to represent the involvement (e.g. subject, author, performer) of a role (e.g. patient, healthcare professional)."

Else:

- If the class selected was a CONSTRAINED_RELATED_PARTY enter the following text in the class Notes field (of the class dialogue): "This Class is provided, for ENTRY.subject_of_information, to identify a person in terms of his or her relationship to the subject_of_care."
- Constrain the class attributes for domain use as follows:
 - Annotate any attributes that require refined descriptions by appending the relevant text into the class description (i.e. within the class Notes field of the class dialogue).
 - To constrain the attribute type create an OCL constraint (using an *oclIsTypeOf(<ISO 21090 data type>)* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.
 - Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.

- To constrain the attribute multiplicity create an OCL constraint (using an *ocllsDefined()* or *ocllsUndefined()* expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to *LRA Invariant*.
 - Set the Status field to *Proposed*.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- To constrain a tagged value create a new tagged value for the class and:
 - Prefix the tag name with the attribute name (e.g. *unit.enum* becomes *value.unit.enum*).
 - Specify the tag value as appropriate.
- Create a realization association between the FUNCTIONAL_ROLE_CHOICE and any CONSTRAINED_FUNCTIONAL_ROLE or Constrained_CR_Participation classes, or between the RELATED_PARTY_CHOICE and any CONSTRAINED_RELATED_PARTY classes. For each realization association:
 - The source class shall be a CONSTRAINED_FUNCTIONAL_ROLE, Constrained_CR_Participation or CONSTRAINED_RELATED_PARTY class.
 - The target class shall be the FUNCTIONAL_ROLE_CHOICE or RELATED_PARTY_CHOICE class.
- Select a CR_Role_Choice class from the LRA Class Templates resource folder:
 - Make the selected class a specialization of the relevant Participation domain model class identified in Step 2 above.
 - In the class Name field (of the class dialogue) enter a short name that describes the class based on its data element content requirements. The name shall be formatted in camel case.
 - Select the Abstract field (of the class dialogue).
 - Set the Status field (of the class dialogue) to *Proposed*.
 - In the class Notes field (of the class dialogue) enter a description of the class based on the provided data element content requirement, commencing:
 - “Represents a choice of CR_Role classes when a CR_Role is being used to describe the details of an entity that participates...”
 - For example,

Class : OtherRoleChoice1

General Details Require Constraints Links Scenario Files

Name: OtherRoleChoice1

Stereotype: CR_Role_Choice ☒ Abstract

Author: Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A x^2 x_2

Represents a choice of CR_Role classes when a CR_Role is being used to describe the details of an entity that participates in a non-specific 'other participation' manner.

- Create an association between the CONSTRAINED_FUNCTIONAL_ROLE, Constrained_CR_Participation or CONSTRAINED_RELATED_ROLE class and the CR_Role_Choice class. For each association:
 - The source class shall be the CONSTRAINED_FUNCTIONAL_ROLE, Constrained_CR_Participation or CONSTRAINED_RELATED_ROLE class.
 - The target class shall be a CR_Role_Choice class.
 - Set the Direction field (in the dependency properties dialogue) to Source -> Destination.
- The CR_Role_Choice class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting Advanced | Make Composite. The class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
- Within the child diagram of the CR_Role_Choice class:

- Select any `Constrained_CR_Role` specialization classes from the Project Browser in accordance with the data element content requirements.
- Create a realization association between the `CR_Role_Choice` class and each `Constrained_CR_Role` specialization class. For each realization association:
 - The source class shall be a `Constrained_CR_Role` specialization class.
 - The target class shall be the `CR_Role_Choice` class.
- Suppress the visibility of all attributes and constraints of each `Constrained_CR_Role` specialization class using the feature visibility (see EA) of the class.
- The `CR_Role` specialization class should be made “composite” (a pointer to a child diagram) by right-clicking on the class in EA and selecting `Advanced | Make Composite`. The class should then be added to the child diagram by dragging that class from the Project Browser. All subsequent alterations to that class should be made in the child diagram. Note when using composite classes, care should be taken in each subsequent step to ensure that additional classes are contained in the correct package (and do not become children of the composite class).
- Within the child diagram of the `Constrained_CR_Role` class:
 - Select the required `Constrained Participations` classes from the LRA Class Templates folder in order to refine the `Participations Domain Model` identified in Step 2 above. `Constrained Participations` classes are identifiable by the prefix “`Constrained_CR_`” with the exception of the *`Constrained_CR_Participation`* class. For each `Constrained Participation` class:
 - Make the selected class a specialization of the relevant `Participation domain model` class identified in Step 2 above.
 - In the class Name field (of the class dialogue) enter a short name that describes the `Constrained Participation` class based on its data element content requirements. The name shall be formatted in camel case.
 - Set the Status field (of the class dialogue) to *Proposed*.
 - In the class Notes field (of the class dialogue) enter a description of the `Constrained Participation` class based on the data element content requirement.
 - Constrain the class attributes for domain use as follows:
 - a. Annotate any attributes that require refined descriptions by appending the relevant text into the class description (i.e. within the class Notes field of the class dialogue).
 - b. To constrain the attribute type create an OCL constraint (using an `oclIsTypeOf(<ISO 21090 data type>)` expression) within the constraints tab of the Class dialogue:

- Set the constraint type (in the Type field) to LRA Invariant.
 - Set the Status field to Proposed.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- c. To constrain the attribute multiplicity create an OCL constraint (using an `oclIsDefined()` or `oclIsUndefined()` expression) within the constraints tab of the Class dialogue:
 - Set the constraint type (in the Type field) to LRA Invariant.
 - Set the Status field to Proposed.
 - Enter the constraint name (in the Constraint field).
 - Enter a descriptive comment and a constraint expression into the main field.
- d. To constrain a tagged value create a new tagged value for the class and:
 - Prefix the tag name with the attribute name (e.g. `unit.enum` becomes `value.unit.enum`).
 - Specify the tag value as appropriate.
- Create any required associations between the Constrained Participation class and any other Constrained Participation classes in order to fulfil the data element content requirements. All associations created should be valid refinements of the Participations Domain Model identified in Step 2 above. For each association:
 - a. Include a relevant description of the association (in the Notes field of the association properties dialogue).
 - b. Within the Source Role tab of the association properties dialogue specify a role name (when appropriate) and multiplicity in accordance with the data element content requirements.
 - c. Within the Target Role tab of the association properties dialogue specify a role name (when appropriate) and multiplicity in accordance with the data element content requirements.

3.3 LRA SNOMED CT Expression Constraints

A SNOMED CT Expression Constraint is a computable rule that can be applied to an instance of a SNOMED CT expression to test whether it complies with rules that may

relate to its meaning and/or compositional structure. There are 2 types of expression constraint:

- Semantic - a type of expression constraint which limits the possible meanings that can be expressed by a field.
- Literal - a type of expression constraint which constrains the ways in which an expression may be constructed to represent a given meaning.

Default expression constraints (of either type) are constraints which are bound to reference model classes and which are therefore inherited by default by conforming domain model classes. An expression constraint bound to a domain model may be further refined but must always conform to the default expression constraint from which it derives.

3.3.1 Semantic Expression Constraints

#TODO: John Arnett / Yongsheng Gao to complete?

3.3.2 Literal Expression Constraints

#TODO: John Arnett / Yongsheng Gao to complete?

4 Collaborative Working Approach

The approach to developing LRA Model Artefacts for LRA Release 2.0 requires collaborative working between members of the LRA knowledge modelling, technical modelling and terminology teams. This section attempts to describe this approach.

4.1 Collaborative Communication Log

A Collaborative Communication Log has been created to enable members of the knowledge modelling, technical modelling and terminology teams to obtain clarification, share ideas and update the modelling status of each data element content requirement.

This log has been created in Enterprise Architect within the *Logical Record Architecture for Health and Social Care.LRA Model Artefacts.Knowledge.LRA Analysis.Data Element Definitions Palette.Candidate Data Elements.cDED Status - <Domain>* package (where <Domain> specifies the relevant knowledge domain).

Within this log each Candidate Data Element Definition specified in the knowledge space will have an entry. Each log entry is modelled within Enterprise Architect as an *Issue*. The *Issues* are viewed as an element list within a Maintenance Diagram as per the screenshot below:

Name	Alias	Status	Difficulty	Priority	Pha...	Ver...	Author	Created	Modified
cded_HaemodialysisTreatmentType	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 08:...	15/10/2009 09:...
cded_OralFluidRestriction	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 08:...	15/10/2009 08:...
cded_HaemodialysisTreatmentFrequency	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	14/10/2009 15:...	15/10/2009 08:...
cded_DurationOnHaemodialysis	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 09:...	15/10/2009 09:...
cded_TargetDryWeight	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 09:...	15/10/2009 09:...
cded_HaemodialysisBloodFlowRate	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 09:...	15/10/2009 09:...
cded_DialyserName	Technical	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 09:...	15/10/2009 09:...
cded_DialysateName	Knowledge	Proposed	Medium	Medium	1.0	1.0	peot1	15/10/2009 09:...	15/10/2009 09:...

Each *Issue* has a series of properties which can be edited in the Issue dialogue (see screenshot below):

Within the LRA Collaborative working approach these properties are used as follows:

Property name	Description
Short Description	The name of the candidate data element definition.
Alias	<p>A free-text field to describe the team with which the current responsibility for this candidate data element definition resides. Possible values are:</p> <ul style="list-style-type: none"> ○ <i>Knowledge</i> ○ <i>Technical</i> ○ <i>Terminology</i> ○ (or a comma separated combination of the above) <p>This field shall only be blank if the value in the status field is <i>Approved</i>.</p>
Status	<p>Describes the current modelling status of this candidate data element definition. The available states are used as follows:</p> <ul style="list-style-type: none"> ○ <i>Approved</i> – all modelling is complete. The value in the Alias field shall be blank. ○ <i>Implemented</i> – technical modelling is complete but checking SNOMED CT expression constraints validity of domain models and the capturing of the fulfilment of knowledge requirements is outstanding. The value in the Alias field shall be <i>Terminology</i> or <i>Knowledge</i>. ○ <i>Mandatory</i> – not used. ○ <i>Proposed</i> – discussion / clarification ongoing. Any allowable value in the Alias field may be used. ○ <i>Validated</i> – not used.
Difficulty	Not used. Auto-populated by EA but ignored.
Priority	Not used. Auto-populated by EA but ignored.
Author	Auto-populated by EA. Do not edit.

Key Words	Not used.
Type	Not used. Auto-populated by EA but ignored.
Phase	Not used. Auto-populated by EA but ignored.
Version	Not used. Auto-populated by EA but ignored.
Last Update	Auto-populated by EA. Do not edit.
Created	Auto-populated by EA. Do not edit.
Notes	<p>A free-text field to capture ongoing analysis comments and requests for further clarification on this candidate data element definition. Each comment is prefixed with the initials of the author of the comment.</p> <p>Once technical modelling is complete (i.e. the value in the Alias and Status fields are <i>Knowledge</i> and <i>Implemented</i> respectively) a comment should be entered to describe the Interface model that fulfils the requirements of this candidate data element definition.</p>

Appendix A: LRA Constraint Types

The following table describes the available LRA Constraint Types:

LRA Constraint Type	Description
Initial value	A state the object must be in at creation time
Invariant	A state the object must always be in
LRA Initial	An initial value constraint specified by the LRA
LRA Invariant	An invariant constraint specified by the LRA
Post-condition	An ending state that must be met
Pre-condition	A starting state that must be met