GDMUML

Genealogical Data Models in UML

GENTECH-GDM

Reference Model

GENEALOGICAL MODELS IN UML

GENTECH-GDM Reference Model

Version 0.1
January 29, 2003

© 2003, Stanley Mitchell
Email: stanlmitchell@yahoo.com

Table of Contents

Section

1

Introduction

A brief statement of purpose; an overview of naming conventions and data types used herein.

The GENTECH Genealogical Data Model was published in May 2000[3]. In its original form, it was presented as an Entity Relationship model[1] for a database system. This model has received attention from the genealogical community as a future framework for the design of systems to aid in genealogical research.

In this document, the GENTECH model is recast in UML terms. All of the original entities are retained as well as their attributes. Additional classes have been added to explicitly model data collections that represent the various types of records stored in a database.

This document is intended to be a reference to the GENTECH Genealogical Data Model; it does not extend the original model with one exception: the addition of a ***Date*** sub-model.

Conventions

A class or data type is denoted with bold italic characters with the first character of each noun or sub-noun, capitalized. For example, ***RepositorySource*** and ***Source*** are class names. A UML stereotype called a **collection** is used to distinguish a class, which represents a generic collection of any type of item. The standard UML notation of a stereotype[2] encloses the name in guillemets, so a ***CitationParts*** **«collection»** is a class, which represents a collection of ***CitationPart*** instances.

When all of the collections defined in the model are combined, they form a *dataset* which may be stored in a database file, an XML file, or another form of persistent storage. Collections are treated as "opaque" objects; their characteristics will depend upon a particular implementation. For this reason, collection objects are defined but their data members are not described.

An entity from the ER model is denoted with bold capitalized characters. If the name is formed from more than one noun, the nouns are separated by hyphens. For example, **REPOSITORY-SOURCE** and **SOURCE** are entity names.

The names of data members are italicized, as in *AscendingDescendingNone*.

The values of ***Enum*** types use a monospace font, as in `Ascending`, `Descending`, or `None`.

Data Types

Some primitive types have been defined as an aid to understanding the usage of class data members. When an ER entity contains a primary database key, the corresponding class contains a "Unique ID" data member. This type is given the shorthand name, ***UID***. A UID is only required to be unique amongst the instances in its class.

When an ER entity contains a foreign database key, the corresponding class contains a pointer type. For example, if the foreign key were a UID of a **SOURCE**, then the corresponding class would have a pointer to a ***Source***. The data member name would be *SourceRef*. A type name followed by an asterisk is the notation used for a pointer to the type. For example, ***Source**** is a pointer to an instance of the ***Source*** class.

In several cases, a data member can be enumerated as one of a discrete set of values. For example, the *AscendingDescendingNone* member can be in one of three possible states. This type is given the shorthand name, ***Enum***.

Boolean data members are those, which can be represented by either True or False.

All textual data is given the type ***String***. A Unicode string is implied.

Integer data members are used for open-ended ranges or sequences. *SequenceNumber* and *Priority* are examples.

An ***Object*** data member refers to a generic type. Here it is used to refer to an external file that contains a bitmapped image or audio data.

Section

2

Administrative Sub-model

The Administrative sub-model defines genealogical projects which use researchers to achieve research objectives through the execution of a set of research activities.

Activity

DEFINITION

An action a researcher takes to accomplish a research objective. Two sub-types are distinguished: ***Search*** and ***AdministrativeTask***. This is a base class to these two sub-classes.

DATA MEMBERS

ActivityID: a ***UID*** that uniquely identifies this instance

ResearcherRef: a pointer to a ***Researcher*** who is assigned the activity

ScheduledDate: pointer to the ***Date*** when the ***Researcher*** will start this activity or a Null pointer if the activity is not scheduled

CompletedDate: pointer to the ***Date*** when the ***Researcher*** completed this activity or a Null pointer if the activity is not completed

TypeCode: an ***Enum***, which is either `Search` or `AdministrativeTask`

Status: an ***Enum***, which indicates the current status of the activity; possible values might include: `Completed`, `OnHold`, `Started`, or `NotStarted`.

Description: a ***String***, which describes what is to be accomplished

Priority: an ***Integer***, that the ***Researcher*** assigns to indicate the ranking order of this ***Activity*** instance relative to other ***Activity*** instances.

Comments: a ***String***, which allows additional information to be entered

EXAMPLE

See examples for ***AdministrativeTask*** and ***Search***.

AdministrativeTask

DEFINITION

This class is derived from the base class, ***Activity***. It does *not* specialize ***Activity*** by adding additional data members. It is used to define activities other than ***Search*** s.

DATA MEMBERS

ActivityID: a ***UID*** that uniquely identifies this instance

ResearcherRef: a pointer to a ***Researcher*** who is assigned the ***AdministrativeTask***

ScheduledDate: pointer to the ***Date*** when the ***Researcher*** will start this activity or a Null pointer if the task is not scheduled

CompletedDate: pointer to the ***Date*** when the ***Researcher*** completed this activity or a Null pointer if the task is not completed

TypeCode: an ***Enum***, which is `AdministrativeTask`

Status: an ***Enum***, which indicates the current status of the ***AdministrativeTask***; possible values might include: `Completed`, `OnHold`, `Started`, or `NotStarted`.

Description: a ***String***, which describes what is to be accomplished

Priority: an ***Integer*** that the ***Researcher*** assigns to indicate the ranking order of this ***Activity*** instance relative to other ***Activity*** instances

Comments: a ***String***, which allows additional information to be entered

EXAMPLE

Data Member	Value
ActivityID	1
ResearchRef	Pointer to ***Researcher*** with *ResearcherID*=1
ScheduledDate	Pointer to ***Date*** instance
CompletedDate	Pointer to ***Date*** instance
TypeCode	`AdministrativeTask`
Status	`NotStarted`
Description	"Update contact information for members of research team."
Priority	100
Comments	"Missing info for Carl Smith – need to call him"

Project

DEFINITION

A project serves to identify the focus of genealogical enquiry. It is up to the ***Researcher*** to define the scope as narrowly or as broadly as needed. Projects have one or more ***Researcher***s. A project may be undertaken on behalf of a client.

DATA MEMBERS

ProjectID: a ***UID*** that uniquely identifies this instance

Name: a ***String***, which holds the name of the project

Description: a ***String***, which provides additional information about the scope of the project

ClientData: a ***String***, which holds the client contact and billing information, if a project is undertaken for a client. The GENTECH-GDM specification[3] suggests that this member could refer to a separate entity that contains client information.

EXAMPLE

Data Member	Value
ProjectID	1
Name	"Ancestors of William J. Sharpe"
Description	"All information about the ancestors of William J. Sharpe of Livingston County, Ky."
ClientData	""

Projects

DEFINITION

A dataset may contain more than one ***Project***. The ***Projects*** **«collection»** contains all of these ***Project*** instances.

COLLECTION ITEM

Project: an instance of ***Project*** which belongs to the dataset.

Researcher

DEFINITION

This class represents a researcher who is responsible for entering data into the system. Each piece of data has a reference to the researcher who contributed it. Data which is imported (e.g., from a GEDCOM file) from work by other researchers needs to be carefully attributed to appropriate authors.

DATA MEMBERS

ResearcherID: a ***UID*** that uniquely identifies this instance

Name: a ***String***, which holds the full name of the researcher

Address: a ***String***, which provides the address of the researcher

PlaceRef: a pointer to a ***Place*** instance which specifies part of the address

Comments: a ***String***, which holds any additional information about the researcher

EXAMPLE

Data Member	Value
ResearcherID	5
Name	"Carl Smith"
Address	"Fremont, CA, USA"
Comments	""

Researchers

DEFINITION

A dataset may contain more than one ***Researcher***. The ***Researchers*** **«collection»** contains all of these ***Researcher*** instances.

COLLECTION ITEM

Researcher: an instance of ***Researcher*** which belongs to the dataset.

ResearchObjective

DEFINITION

The goals of a research project can be broken down into a series of objectives. Each of these is represented as an instance of the ***ResearchObjective*** class.

DATA MEMBERS

ResearchObjectiveID: a ***UID*** that uniquely identifies this instance

ProjectRef: a pointer to the ***Project*** to which this ***ResearchObjective*** belongs

Name: a ***String***, which holds a name for the objective

Description: a ***String***, which provides additional details about the objective

SequenceNumber: an ***Integer*** that the researcher assigns to specify the display order of this objective relative to other objectives.

Priority: an ***Integer*** that the researcher assigns to indicate the ranking order of this objective relative to other objectives.

Status: an ***Enum***, which indicates the current status of the ***ResearchObjective***, possible values might include: `Open` or `Closed`.

EXAMPLE

Data Member	Value
ResearchObjectiveID	2
ProjectRef	Pointer to ***Project*** instance with *ProjectID*=1
Name	"Document William J. Sharpe's mother"
Description	"William J. Sharpe's mother is Delilah Neal. Document her relationship to William J. Sharpe. Determine her parents and birthplace."
SequenceNumber	3
Priority	10
Status	`Open`

ResearchObjectives

DEFINITION

A ***Project*** may contain more than one ***ResearchObjective***. A ***ResearchObjectives*** **«collection»** contains all of these ***ResearchObjective*** instances. This collection is owned by the ***Project***.

COLLECTION ITEM

ResearchObjective: an instance of ***ResearchObjective*** which belongs to the ***Project***.

ResearchObjectiveActivitys

DEFINITION

Each ***ResearchObjective*** will require one or more ***Activity*** instances to be completed for the objective to be accomplished. This class represents this collection of ***Activity*** instances. This collection is owned by the ***ResearchObjective***.

COLLECTION ITEM

ResearchObjectiveRef: a pointer to the ***ResearchObjective*** to which this ***Activity*** belongs. All items in a collection will reference the same ***ResearchObjective***.

ActivityRef: a pointer to an ***Activity*** which helps achieve the ***ResearchObjective***

EXAMPLE

Collection Item Data Member	Value
ResearchObjectiveRef	Pointer to ***ResearchObjective*** instancewith *ResearchObjectiveID* =1
ActivityRef	Pointer to ***Activity*** instance with *ActivityID*=1

ResearcherProjects

DEFINITION

This class represents the collection of ***Project***s in which a ***Researcher*** participates.

COLLECTION ITEM

ResearcherRef: a pointer to the ***Researcher*** which participates in a ***Project***. All items in a collection will reference the same ***Researcher***.

ProjectRef: a pointer to a ***Project*** in which the ***Researcher*** participates.

Role: a ***String***, which describes the role the researcher plays in this ***Project***

EXAMPLE

Collection Item Data Member	Value
ResearcherRef	Pointer to ***Researcher*** instance with *ResearcherID* =1
ProjectRef	Pointer to ***Project*** instance with *ProjectID*=1
Role	"Administrator"

Search

DEFINITION

This class is derived from the base class, ***Activity***. It specializes ***Activity*** by adding additional data members to specify the ***Repository***, ***Source***, and the information sought.

DATA MEMBERS

ActivityID: a ***UID*** that uniquely identifies this instance

ResearcherRef: a pointer to a ***Researcher*** who is assigned the ***Search***

ScheduledDate: pointer to the ***Date*** instance when the researcher will start this search or a Null pointer if the search is not scheduled

CompletedDate: pointer to the ***Date*** instance when the researcher completed this search or a Null pointer if the search is not completed

TypeCode: an ***Enum***, which is either `Search` or `AdministrativeTask`

Status: an ***Enum***, which indicates the current status of the ***Search***, possible values might include: `Completed`, `OnHold`, `Started`, or `NotStarted`.

Description: a ***String***, which describes what is to be accomplished

Priority: an ***Integer*** that the researcher assigns to indicate the ranking order of this ***Activity*** relative to other ***Activity*** s

Comments: a ***String***, which allows additional information to be entered

SourceRef: a pointer to a ***Source*** instance, which identifies the material to be examined. Note that *SourceRef* points to an appropriate level of a multi-level ***Source***. If a ***Source*** is being accessed for the first time, it may be sufficient to use a single-level ***Source*** with a single ***CitationParts*** **«collection»** that contains the equivalent of a bibliographic entry.

RepositoryRef: a pointer to a ***Repository,*** where the search is conducted

SearchedFor: a ***String***, which specifies the information sought

EXAMPLE

Data Member	Value
ActivityID	00000002
ResearchRef	Pointer to ***Researcher*** instancewith *ResearcherID*=2
ScheduledDate	Pointer to ***Date*** instance
CompletedDate	Pointer to ***Date*** instance
TypeCode	`Search`
Status	`Started`
Description	"Search 1850 Pope County, Illinois census for Wentzel, Wetzel, Whetsel surnames."
Priority	20
Comments	"Borrow Pope County CD from Cindy"
SourceRef	Pointer to ***Source*** instance for 1850 Pope County, Illinois Federal Census
RepositoryRef	Pointer to ***Repository*** instance for Heritage Quest CD-ROM
SearcherFor	"Wentzel, Wetzel, Whetsel"

SourceGroup

DEFINITION

Each instance of this class represents a tag-name that can be applied to a group of ***Source*** s.

DATA MEMBERS

SourceGroupID: a ***UID*** that uniquely identifies this instance

SourceGroupName: a ***String***, which is the tag-name assigned to a group of ***Source*** s

EXAMPLE

Data Member	Value
SourceGroupID	4
SourceGroupName	"Tombstone"

SourceGroups

DEFINITION

A dataset may contain more than one ***SourceGroup***. The ***SourceGroups*** **«collection»** contains all of these ***SourceGroup*** instances.

COLLECTION ITEM

SourceGroup: an instance of ***SourceGroup*** which belongs to the dataset.

SourceGroupSource

DEFINITION

This collection holds references to ***Source*** instances which share the same ***SourceGroup*** tag-name. There will be separate collections for each defined ***SourceGroup***. These collections are owned by the dataset.

COLLECTION ITEM

SourceGroupRef: a pointer to a ***SourceGroup*** instance. All items in a collection will reference the same ***SourceGroup*** instance.

SourceRef: a pointer to a ***Source*** which is a member of the ***SourceGroup***.

SuretyScheme

DEFINITION

Each ***Project*** uses a ***SuretyScheme*** to assign levels of certainty to ***Assertion*** instances. Each ***Project*** uses at most one ***SuretyScheme***.

DATA MEMBERS

SuretySchemeID: a ***UID*** that uniquely identifies this instance

SuretySchemeName: a ***String***, which is the name assigned to this ***SuretyScheme***.

SuretySchemeDescription: a ***String***, which is a general description of the ***SuretyScheme***.

EXAMPLE

Data Member	Value
SuretySchemeID	1
SuretySchemeName	"GEDCOM 5.5"
SuretySchemeDescription	"QUAY values used by GEDCOM version 5.5"

SuretySchemes

DEFINITION

A dataset may have more than one ***SuretyScheme*** if it contains more than one ***Project***. This collection contains all of the ***SuretyScheme*** instances used by the dataset.

COLLECTION ITEM

SuretyScheme: an instance of ***SuretyScheme*** which belongs to the dataset.

SuretySchemePart

DEFINITION

This class represents a single certainty level in a ***SuretyScheme***.

DATA MEMBERS

SuretySchemePartID: a ***UID*** that uniquely identifies this instance

SuretySchemeRef: a pointer to a ***SuretyScheme*** instance, to which this part belongs.

SuretySchemePartName: a ***String***, which is the name of the ***SuretySchemePart***.

SuretySchemePartDescription: a ***String***, which is a general description of the ***SuretySchemePart***.

SequenceNumber: an ***Integer*** that is assigned to a ***SuretySchemePart*** that is used to sort the most reliable with the highest numeric values.

EXAMPLE

Data Member	Value
SuretySchemePartID	4
SuretySchemeRef	Pointer to ***SuretyScheme*** with *SuretySchemeID*=1
SuretySchemePartName	"3"
SuretySchemePartDescription	"Direct and primary evidence used, or by dominance of the evidence."
SequenceNumber	4

SuretySchemeParts

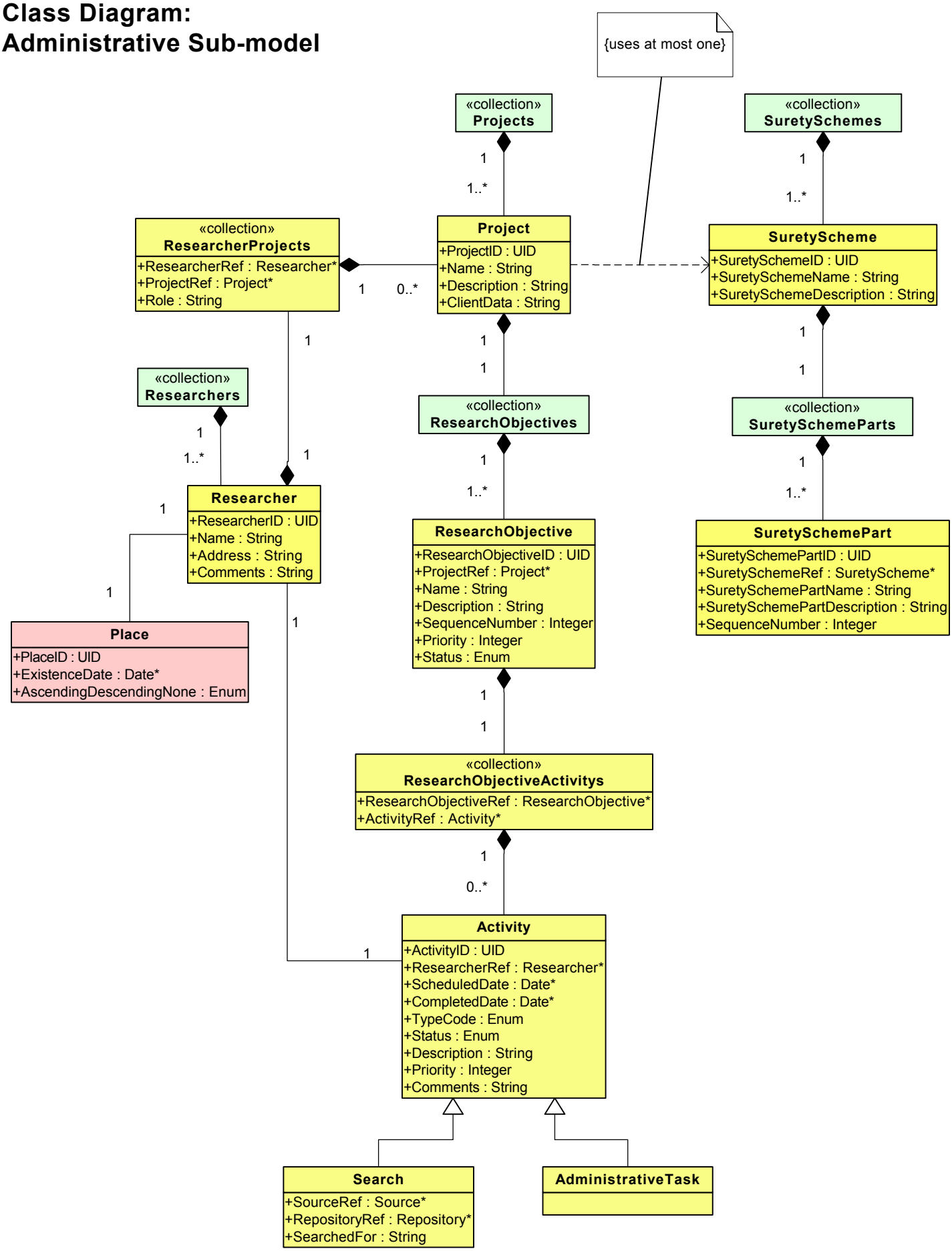
DEFINITION

A dataset may contain more than one ***SuretyScheme***. A ***SuretySchemeParts*** **«collection»** will exist for each ***SuretyScheme***. The ***SuretySchemeParts*** **«collection»**s are owned by the dataset.

COLLECTION ITEM

SuretySchemePart: an instance of ***SuretySchemePart*** which belongs to the ***SuretyScheme***.

Class Diagram:
Administrative Sub-model



Section

3

Evidence Sub-model

Data may come into the system from quite varied sources. Classes of the Evidence sub-model record where this data originated.

CitationPart

DEFINITION

Each instance of this class represents a part of a citation for a ***Source***, such as the author, title, or publication place.

DATA MEMBERS

SourceRef: a pointer to the ***Source*** instance to which this ***CitationPart*** refers

CitationPartTypeRef: a pointer to a ***CitationPartType*** instance to which this ***CitationPart*** refers

CitationPartValue: a ***String***, which contains the citation component

EXAMPLE

Data Member	Value
SourceRef	Pointer to ***Source*** with *SourceID*=3
CitationPartTypeRef	Pointer to ***CitationPartType*** with *CitationPartTypeID*=1
CitationPartValue	"Smith, Thomas D."

CitationParts

DEFINITION

This collection is owned by a single ***Source***. It contains all of the ***CitationPart***s needed for a complete citation for a ***Source*** level. Each of the ***Source***s in a multi-level ***Source***, will have separate ***CitationParts*** **«collection»**s.

COLLECTION ITEM

CitationPart: instances of ***CitationPart*** which belong to a single ***Source*** instance.

CitationPartType

DEFINITION

Each instance of this class represents a type of component in a bibliographic citation or source note. Examples include "Author", "Title", "Volume", etc.

DATA MEMBERS

CitationPartTypeID: a ***UID*** that uniquely identifies this instance

CitationPartTypeName: a ***String***, which is the name assigned to this component

EXAMPLE

Data Member	Value
CitationPartTypeID	1
CitationPartTypeName	"Author"

CitationPartTypes

DEFINITION

A dataset will use many ***CitationPartType***s in its ***CitationParts*** **«collection»**s. This collection contains all of the ***CitationPartType***s used by the dataset.

COLLECTION ITEM

CitationPartType: an instance of ***CitationPartType*** which belongs to the dataset.

Repository

DEFINITION

Each instance of this class represents the location of a ***Source***.

DATA MEMBERS

RepositoryID: a ***UID*** that uniquely identifies this instance

PlaceRef: a pointer to a ***Place*** instance which specifies the location

Name: a ***String***, which is the name given to the ***Repository***

Address: a ***String***, which is the current address of the ***Repository***

Phone: a ***String***, which lists the phone numbers for contacting the ***Repository***

Hours: a ***String***, which lists the hours during which the ***Repository*** is open for access

Comments: a ***String***, which contains additional information about the ***Repository*** - a contact person, for example

EXAMPLE

Data Member	Value
RepositoryID	1
PlaceRef	Pointer to ***Place*** instance with *PlaceID*=23
Name	"Pacific Region's NARA"
Address	"1000 Commodore Drive, San Bruno, CA 94066-2350"
Phone	"650-876-9009"
Hours	"Monday through Friday, 7:30 A.M. to 4:00 P.M. Wednesday, 4 P.M. to 8 P.M. (Microfilm research only)"
Comments	"Self-service microfilm readers and reader-printers"

Repositorys

DEFINITION

A dataset will use many instances of ***Repository*** to record ***Search****s* for and locations of ***Source****s*. This collection contains all of the ***Repository****s* referenced by the dataset.

COLLECTION ITEM

Repository: an instance of ***Repository*** which belongs to the dataset.

RepositorySource

DEFINITION

Each instance of this class represents the occurrence of a specific ***Source*** at a ***Repository***. Finding a ***Source*** and ***Repository*** combination is the result of a ***Search***.

DATA MEMBERS

RepositoryRef: pointer to a ***Repository*** instance

SourceRef: pointer to a ***Source*** instance

ActivityRef: pointer to a ***Search*** instance

CallNumber: a ***String***, containing the unique call number for the ***Source***, at the specified repository.

Description: a ***String***, containing notes about the particular ***Source***; its condition for example.

EXAMPLE

Data Member	Value
RepositoryRef	Pointer to a ***Repository*** instance with *RepositoryID*=1
SourceRef	Pointer to a ***Source*** instance with *SourceID*=12
ActivityRef	Pointer to an ***Search*** instance with *ActivityID*=3
CallNumber	""
Description	"Single microfilm copy of Micropublication M593, Role 482"

RepositorySources

DEFINITION

This collection contains all of the ***RepositorySource*** instances referenced by the dataset.

COLLECTION ITEM

RepositorySource: an instance of ***RepositorySource*** which belongs to the dataset.

Representation

DEFINITION

Each instance of this class corresponds to a representation of a ***Source***. If the representation is plain text or a form of electronic media, then a reference to it can be stored with the class instance. Otherwise, a *PhysicalFileCode* can refer to a representation which is physically filed or stored outside of the dataset.

DATA MEMBERS

SourceRef: pointer to a ***Source*** instance

RepresentationTypeRef: pointer to a ***RepresentationType*** instance

PhysicalFileCode: a ***String***, which is a filing code or other reference number for locating a ***Representation*** when it is external to the dataset.

Medium: a ***String***, which is the medium type of the ***Representation***. Some examples include paper (for a deed), stone (for a tombstone), sound (for a voice recording), etc.

Content: an ***Object***, which holds the actual content of the ***Representation***. Some examples include a jpeg image of a census page, a digital camera image of a tombstone, or a wav file of a voice recording.

Comments: a ***String***, which holds additional notes about the ***Representation***

EXAMPLE

Data Member	Value
SourceRef	Pointer to a ***Source*** instance with *SourceID*=12
RepresentationTypeRef	Pointer to an ***RepresentationType*** instance
PhysicalFileCode	“”
Medium	“paper”
Content	M593_482_175A.jpg
Comments	“Scanned image of census page 175A”

Representations

DEFINITION

This collection contains all of the ***Representation*** instances referenced by the dataset.

COLLECTION ITEM

Representation: an instance of ***Representation*** which belongs to the dataset.

RepresentationType

DEFINITION

Each instance of this class specifies a type of ***Representation***. Types could include "Text", "JPEG Bitmap", "WAV Sound File", etc.

DATA MEMBERS

RepresentationTypeID: a ***UID*** that uniquely identifies this instance

RepresentationTypeName: a ***String***, which is the ***Representation*** type name

EXAMPLE

Data Member	Value
RepresentationTypeID	1
RepresentationTypeName	"Text"

RepresentationTypes

DEFINITION

This collection contains all of the ***RepresentationType*** instances referenced by the dataset.

COLLECTION ITEM

RepresentationType: an instance of ***RepresentationType*** which belongs to the dataset.

Source

DEFINITION

Each instance of this class represents a source of data. A source may be represented by more than one level of detail. This allows the information contained in a bibliographic entry to be represented as a higher level ***Source*** instance and for source notes to be represented by multiple lower-level ***Source*** instances that reference the common higher level ***Source***.

DATA MEMBERS

SourceID: a ***UID*** that uniquely identifies this instance

HigherSourceRef: a pointer to a higher level ***Source*** instance or a Null pointer if none exists.

SubjectPlaceRef: a pointer to a ***Place*** instance which identifies the place of the subject of the ***Source***

JurisdictionPlaceRef: a pointer to a ***Place*** instance which identifies the place of the jurisdiction of the ***Source***

ResearcherRef: a pointer to a ***Researcher*** instance which represents the contributor of the ***Source*** data.

SubjectDate: a pointer to a ***Date*** instance which specifies the date of the subject of the ***Source***.

Comments: a ***String***, which supplies additional information about the ***Source***, such as its condition or readability.

EXAMPLE

Data Member	Value
SourceID	23
HigherSourceRef	Null
SubjectPlaceRef	Pointer to ***Place*** instance with *PlaceID*=3
JurisdictionPlaceRef	Pointer to ***Place*** instance with *PlaceID*=4
ResearcherRef	Pointer to ***Researcher*** instance with *ResearcherID*=1
SubjectDate	Pointer to ***Date*** instance
Comments	"A deed to William C. Sharp may exist in these records"

Sources

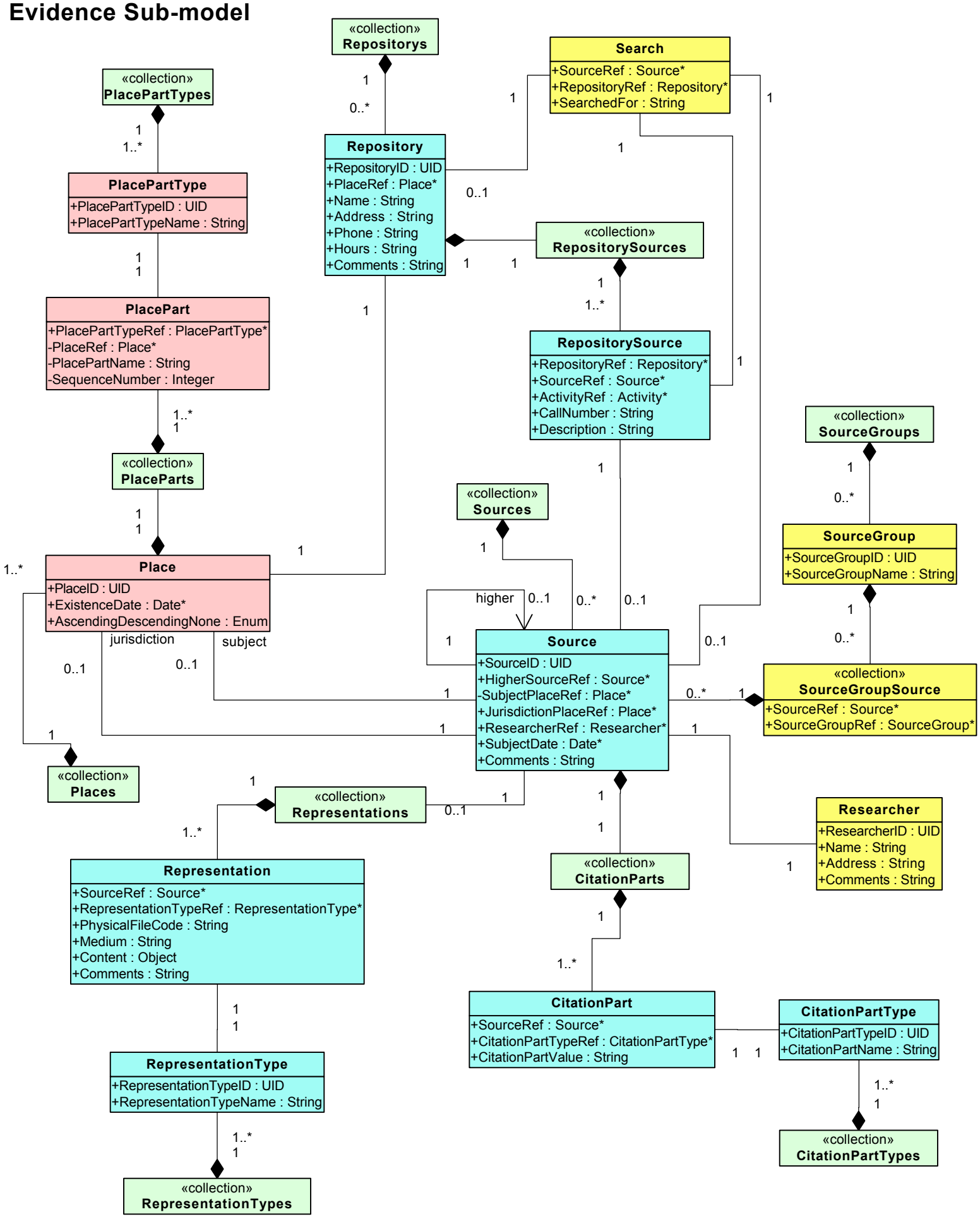
DEFINITION

This collection contains all of the ***Source*** instances referenced by the dataset.

COLLECTION ITEM

Source: an instance of ***Source*** which belongs to the dataset.

Class Diagram: Evidence Sub-model



Section

4

Conclusional Sub-model

From evidence come conclusions. The classes of the Conclusional sub-model formalize the formation of assertions from evidence.

Assertion

DEFINITION

This class represents a single assertion statement made by a ***Researcher***. An ***Assertion*** may be derived from a lowest level ***Source*** or it may be derived from a collection of other ***Assertion***s.

DATA MEMBERS

AssertionID: a ***UID*** that uniquely identifies this instance

SuretySchemePartRef: a pointer to a ***SuretySchemePart*** instance that indicates the researcher's certainty in this assertion.

ResearcherRef: a pointer to a ***Researcher*** instance that is responsible for this ***Assertion***.

SourceRef: a pointer to a ***Source*** instance which identifies the direct evidence that this ***Assertion*** is based upon; this member will be a Null pointer, if this ***Assertion*** is derived from other ***Assertion***s.

Subject1Type: an ***Enum***, which can be one of: `Persona`, `Event`, `Characteristic`, or `Group`

Subject1Ref: a pointer to a ***Persona***, ***Event***, ***Characteristic***, or ***Group*** instance which specifies the first subject of the ***Assertion*** statement.

Subject2Type: an ***Enum***, which can be one of: `Persona`, `Event`, `Characteristic`, or `Group`

Subject2Ref: a pointer to a ***Persona***, ***Event***, ***Characteristic***, or ***Group*** instance which specifies the second subject of the ***Assertion*** statement.

Value: a ***String***, which is the "value" of the ***Assertion*** statement

Rationale: a ***String***, which is explains the ***Researcher's*** basis for the ***Assertion***

Disproved?: a ***Boolean***, which if True, indicates the ***Researcher*** believes this ***Assertion*** is no longer valid

EXAMPLE

Data Member	Value
AssertionID	23
SuretySchemePartRef	Pointer to ***SuretySchemePart*** instance with *SuretySchemePartID*=4
ResearcherRef	Pointer to ***Researcher*** instance with *ResearcherID*=1
SourceRef	Pointer to ***Source*** instance with *SourceID*=3
Subject1Type	`Persona`
Subject1Ref	Pointer to ***Persona*** instance with *PersonaID*=2
Subject2Type	`Characteristic`
Subject2Ref	Pointer to ***Characteristic*** instance with *CharacteristicID*=5
Value	"carpenter"
Rationale	"William reported in this census that 'carpenter' was his occupation."
Disproved?	false

Assertions

DEFINITION

This collection contains all of the ***Assertion*** instances referenced by the dataset.

COLLECTION ITEM

Assertion: an instance of ***Assertion*** which belongs to the dataset.

AssertionAssertion

DEFINITION

This class represents the connection between one lower level ***Assertion*** and a higher level ***Assertion***.

DATA MEMBERS

AssertionRefLow: a pointer to a lower level ***Assertion*** instance

AssertionRefHigh: a pointer to a derived higher level ***Assertion*** instance.

SequenceNumber: an ***Integer*** that the ***Researcher*** assigns to specify the order of the lower level ***Assertion***s that are combined to form the single higher level ***Assertion***.

EXAMPLE

Data Member	Value
AssertionRefLow	Pointer to ***Assertion*** instance with *AssertionID*=4
AssertionRefHigh	Pointer to ***Assertion*** instance with *AssertionID*=15
SequenceNumber	3

AssertionAssertions

DEFINITION

This collection contains all of the ***AssertionAssertion*** instances that support a derived higher level ***Assertion***, i.e. all items in the collection reference the same higher level assertion, *AssertionRefHigh*. These collections are owned by the dataset.

COLLECTION ITEM

AssertionAssertion: an instance of ***AssertionAssertion*** which references a specific derived ***Assertion***.

Characteristic

DEFINITION

This class represents a subject type for an ***Assertion*** statement.

DATA MEMBERS

CharacteristicID: a ***UID*** that uniquely identifies this instance

PlaceRef: a pointer to a ***Place*** instance associated with this ***Characteristic***

CharacteristicDate: a pointer to a ***Date*** instance associated with this ***Characteristic***

AscendingDescendingNone: an ***Enum***, which may assume the values: `Ascending`, `Descending`, or `None`. It specifies the sort order of items in the ***CharacteristicParts*** **«collection»**.

EXAMPLE

Data Member	Value
CharacteristicID	12
PlaceRef	Pointer to ***Place*** instance with *PlaceID*=13
CharacteristicDate	Pointer to a ***Date*** instance
AscendingDescendingNone	`Ascending`

Characteristics

DEFINITION

This collection contains all of the ***Characteristic*** instances that are referenced by the dataset.

COLLECTION ITEM

Characteristic: an instance of ***Characteristic*** which is the subject of an ***Assertion***.

CharacteristicPart

DEFINITION

This class represents an attribute which describes an individual, such as their occupation, name, or hair color. It holds the "value" of the part.

DATA MEMBERS

CharacteristicPartID: a ***UID*** that uniquely identifies this instance

CharacteristicRef: a pointer to a ***Characteristic*** instance associated with this ***CharacteristicPart***

CharacteristicPartTypeRef: a pointer to a ***CharacteristicPartType*** instance associated with this ***CharacteristicPart***

CharacteristicPartName: a ***String***, which is the ***CharacteristicPart*** value.

SequenceNumber: an ***Integer*** that specifies the order of the ***CharacteristicPart****s* in the ***CharacteristicParts*** **«collection»**.

EXAMPLE

Data Member	Value
CharacteristicPartID	10
CharacteristicRef	Pointer to the ***Characteristic*** instance with *CharacteristicID*=7
CharacteristicPartTypeRef	Pointer to the ***CharacteristicPartType*** instance with *CharacteristicPartTypeID*=12
CharacteristicPartName	"Carpenter"
SequenceNumber	1

CharacteristicParts

DEFINITION

This collection contains all of the ***CharacteristicPart*** instances that are referenced by a ***Characteristic***.

COLLECTION ITEM

CharacteristicPart: an instance of ***CharacteristicPart*** which is referenced by a ***Characteristic***.

CharacteristicPartType

DEFINITION

This class represents an attribute which describes an individual, such as occupation, name, or hair color. It holds the "type" of the part.

DATA MEMBERS

CharacteristicPartTypeID: a ***UID*** that uniquely identifies this instance

CharacteristicPartTypeName: a ***String***, which is the ***CharacteristicPartType*** name

EXAMPLE

Data Member	Value
CharacteristicPartTypeID	12
CharacteristicPartTypeName	"Occupation"

CharacteristicPartTypes

DEFINITION

This collection contains all of the ***CharacteristicPartType*** instances that are referenced by ***CharacteristicPart*** instances. It is owned by the dataset..

COLLECTION ITEM

CharacteristicPartType: an instance of ***CharacteristicPartType*** which is referenced by a ***CharacteristicPart***.

Event

DEFINITION

This class represents a subject type for an ***Assertion*** statement.

DATA MEMBERS

EventID: a ***UID*** that uniquely identifies this instance

EventTypeRef: a pointer to an instance of ***EventType***, that specifies the type of event.

PlaceRef: a pointer to a ***Place*** instance associated with this ***Event***

EventName: a ***String***, which is the ***Event*** name

EventDate: a pointer to a ***Date*** instance associated with this ***Event***

EXAMPLE

Data Member	Value
EventID	10
EventTypeRef	Pointer to the ***EventType*** instance with *EventTypeID*=7
PlaceRef	Pointer to the ***Place*** instance with *PlaceID*=12
EventName	"Marriage of William C. Sharp and Delilah Neal"
EventDate	Pointer to a ***Date*** instance

Events

DEFINITION

This collection contains all of the ***Event*** instances that are referenced by the dataset.

COLLECTION ITEM

Event: an instance of ***Event*** which is the subject of an ***Assertion***.

EventType

DEFINITION

This class represents the type of an event referenced by an ***Event*** in an ***Assertion*** statement.

DATA MEMBERS

EventTypeID: a ***UID*** that uniquely identifies this instance

EventTypeName: a ***String***, which is the ***EventType*** name

EXAMPLE

Data Member	Value
EventTypeID	13
EventTypeName	"Marriage"

EventTypes

DEFINITION

This collection contains all of the ***EventType*** instances that are referenced by ***Event***s in the dataset.

COLLECTION ITEM

EventType: an instance of ***EventType*** which is associated with an ***Event***.

EventTypeRole

DEFINITION

This class represents the role an individual plays in an event.

DATA MEMBERS

EventTypeRoleID: a ***UID*** that uniquely identifies this instance

EventTypeRef: a pointer to an instance of ***EventType***.

EventTypeRoleName: a ***String***, which is the ***EventTypeRole*** name

EXAMPLE

Data Member	Value
EventTypeRoleID	3
EventTypeRef	Pointer to ***EventType*** instance,
EventTypeRoleName	"groom"

EventTypeRoles

DEFINITION

This collection contains all of the ***EventTypeRole*** instances that are referenced by ***Event***s in the dataset.

COLLECTION ITEM

EventTypeRole: an instance of ***EventTypeRole*** which is associated with an ***Event***.

Group

DEFINITION

This class represents a subject type for an ***Assertion*** statement.

DATA MEMBERS

GroupID: a ***UID*** that uniquely identifies this instance

GroupTypeRef: a pointer to an instance of ***GroupType***

PlaceRef: a pointer to a ***Place*** instance associated with this ***Group***

GroupName: a ***String***, which is the ***Group*** name

GroupDate: a pointer to a ***Date*** instance associated with this ***Group***

GroupCriteria: a ***String***, which describes the criteria for admission to the ***Group***

EXAMPLE

Data Member	Value
GroupID	10
GroupTypeRef	Pointer to the ***GroupType*** instance with *GroupTypeID*=7
PlaceRef	Pointer to the ***Place*** instance with *PlaceID*=12
GroupName	"Neighbors of William C. Sharp"
GroupDate	Pointer to a ***Date*** instance
GroupCriteria	"Neighbors of William C. Sharp"

Groups

DEFINITION

This collection contains all of the ***Group*** instances that are referenced by the dataset.

COLLECTION ITEM

Group: an instance of ***Group*** which is the subject of an ***Assertion***.

GroupType

DEFINITION

This class represents the type of a group referenced by a ***Group*** in an ***Assertion*** statement.

DATA MEMBERS

GroupTypeID: a ***UID*** that uniquely identifies this instance

GroupTypeName: a ***String***, which is the ***GroupType*** name

AscendingDescendingNone: an ***Enum***, which may assume the values: `Ascending`, `Descending`, or `None`. It specifies the sort order for members of the ***Group***.

EXAMPLE

Data Member	Value
GroupTypeID	13
GroupTypeName	"Neighbors Occupying Contiguous Property"
AscendingDescendingNone	`Ascending`

GroupTypes

DEFINITION

This collection contains all of the ***GroupType*** instances that are referenced by ***Groups*** in the dataset.

COLLECTION ITEM

GroupType: an instance of ***GroupType*** which is associated with a ***Group***.

GroupTypeRole

DEFINITION

This class represents the role an individual plays in a group.

DATA MEMBERS

GroupTypeRoleID: a ***UID*** that uniquely identifies this instance

GroupTypeRef: a pointer to an instance of ***GroupType***

GroupTypeRoleName: a ***String***, which is the ***GroupTypeRole*** name

SequenceNumber: an ***Integer*** that specifies the order of the members in the ***Group***.

EXAMPLE

Data Member	Value
GroupTypeRoleID	7
GroupTypeRef	Pointer to ***GroupType*** instance
GroupTypeRoleName	"neighbor"
SequenceNumber	2

GroupTypeRoles

DEFINITION

This collection contains all of the ***GroupTypeRole*** instances that are referenced by ***Group***s in the dataset.

COLLECTION ITEM

GroupTypeRole: an instance of ***GroupTypeRole*** which is associated with a ***Group***.

Persona

DEFINITION

This class represents a subject type for an ***Assertion*** statement. It contains the "core identification for each individual in genealogical data"[3].

DATA MEMBERS

PersonaID: a ***UID*** that uniquely identifies this instance

PersonaName: a ***String***, which is the ***Persona's*** name

DescriptionComments: a ***String***, which may contain additional information to help distinguish the person.

EXAMPLE

Data Member	Value
PersonaID	7
PersonaName	"William C. Sharp"
DescriptionComments	"The one born in 1812"

Personas

DEFINITION

This collection contains all of the ***Persona*** instances that are referenced by the dataset.

COLLECTION ITEM

Persona: an instance of ***Persona*** which is the subject of an ***Assertion***.

Place

DEFINITION

This class represents a geographic location. It contains a collection of ***PlaceParts*** that describe the hierarchical relationship of this place to other places.

DATA MEMBERS

PlaceID: a ***UID*** that uniquely identifies this instance

ExistenceDate: a pointer to a ***Date*** instance associated with this ***Place***

AscendingDescendingNone: an ***Enum***, which may assume the values: `Ascending`, `Descending`, or `None`. It specifies the sort order for members of the ***PlaceParts*** **«collection»**.

EXAMPLE

Data Member	Value
PlaceID	13
ExistenceDate	Pointer to a ***Date*** instance
AscendingDescendingNone	`Ascending`

Places

DEFINITION

This collection contains all of the ***Place*** instances that are referenced by the dataset.

COLLECTION ITEM

Place: an instance of ***Place*** which is referenced in the dataset.

PlacePart

DEFINITION

This class contains an element of the collection of ***PlaceParts*** that form a hierarchical description of a place.

DATA MEMBERS

PlacePartID: a ***UID*** that uniquely identifies this instance

PlaceRef: a pointer to a ***Place*** instance.

PlacePartName: a ***String***, which is the ***PlacePart*** name

SequenceNumber: an ***Integer*** that specifies the order of the ***PlaceParts*** in the collection.

EXAMPLE

Data Member	Value
PlacePartID	13
PlaceRef	Pointer to ***Place*** instance
PlacePartName	"Illinois"
SequenceNumber	2

PlaceParts

DEFINITION

This collection contains all of the ***PlacePart*** instances that are referenced by a ***Place***. A ***Place*** instance owns this collection.

COLLECTION ITEM

PlacePart: an instance of ***PlacePart*** which is referenced by a ***Place***.

PlacePartType

DEFINITION

This class contains an item in the collection of ***PlacePartType***s. Each ***PlacePart*** contains a reference to an instance of ***PlacePartType***, which specifies the type of the "part", such as 'Country', 'State', 'City', etc.

DATA MEMBERS

PlacePartTypeID: a ***UID*** that uniquely identifies this instance

PlacePartTypeName: a ***String***, which is the ***PlacePartType*** name

EXAMPLE

Data Member	Value
PlacePartTypeID	13
PlacePartTypeName	"State"

PlacePartTypes

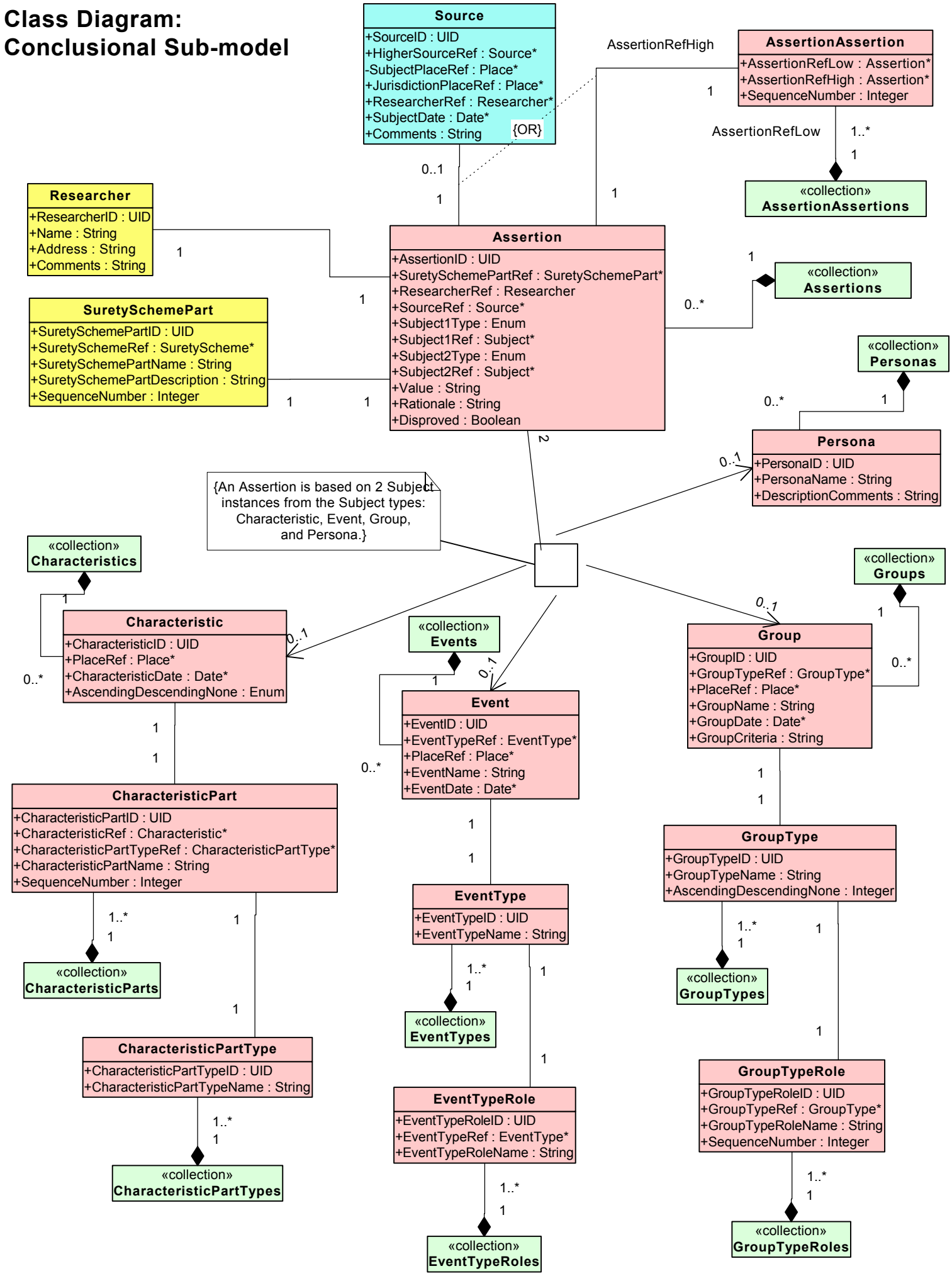
DEFINITION

This collection contains all of the ***PlacePartType*** instances that are referenced by a dataset.

COLLECTION ITEM

PlacePartType: an instance of ***PlacePartType*** which is referenced by a dataset.

Class Diagram: Conclusional Sub-model



Section

5

Date Sub-model

The timeline of events in a person's life is key to genealogical research. A representation of dates is essential for recording exact dates as well as relative and approximate dates.

Date

DEFINITION

This class represents a point date, a date range, or a relative date. Dates are normalized and conversions between different calendar systems are performed using the calendrical calculation algorithms developed by Dershowitz and Reingold[4]

DATA MEMBERS

DateID: a ***UID*** that uniquely identifies this instance

CalendarRef: a pointer to a ***Calendar*** instance, which indicates which calendar system this date was recorded under.

Type: a ***Enum***, which indicates the type of date which this instance records. It can be one of: `Point`, `Range`, `Relative`, `Point-in-Range`, or `Open-Ended-Range`.

- `Point` – a single date component is represented by the *Start* member; e.g. *Start* = {December 12, 1812}.
- `Range` – a range of dates represented by the *Start* and *End* members; e.g. *Start* = {August 27, 1902} to *End* = {January 3, 1903}.
- `Relative` – a single date component is represented by the *Start* member along with *BeforeAfter* and optionally, *Duration*, *DurationPartTypeRef*, and *EventRef*; e.g. *Start* = {August 6, 1855}, *BeforeAfter* = {Before}, encodes "before August 6, 1855".

- `Point-in-Range` – a single date within a range; *Start* = {June 1, 1881} to *End* = {July 25, 1881}, encodes a `Point` date within this range.

- `Open-Ended-Range` – a single date represented by the *Start* member along with *BeforeAfter*, e.g. *Start* = {September 12, 1902} and *BeforeAfter* = {`After`}, encodes "September 12, 1902 to the present".

Start: a ***DateParts*** **«collection»** which defines a starting date for a `Range` or a single `Point` date

NormalizedStartDate: an ***Integer***, the start date expressed as the sequential day number of *Start* since January 1, Gregorian year 1

End: a ***DateParts*** **«collection»** which defines a ending date for a range. This collection is empty for a `Point` ***Date***.

NormalizedEndDate: an ***Integer***, the end date expressed as the sequential day number of *End* since January 1, Gregorian year 1

BeforeAfter: an ***Enum*** which indicates whether a `Relative` date is before or after *Start*.

Duration: an ***Integer*** which specifies the number of units before or after a given start date.

DurationPartTypeRef: a pointer to a ***DatePartType*** which specifies the unit of *Duration*

EventRef: a pointer to a ***Event*** instance, which serves as an anchor for a relative date.

Certainty: an ***Enum*** which indicates the type of measure used to express accuracy. It can be one of: `About, Calculated,` or `Exact`.

EXAMPLE

Data Member	Value
DateID	10
CalendarRef	Pointer to the ***Calendar*** instance with *CalendarID*=1
Type	`Point`
Start	Pointer to ***DataParts*** collection {Month=November, Day=12, Year=1945}
NormalizedStartDate	710347
End	(Not used)
NormalizedEndDate	(Not used)
BeforeAfter	(Not used)
Duration	(Not used)
DurationPartTypeRef	Null
EventRef	Null
Certainty	`Exact`

Dates

DEFINITION

This collection contains all of the ***Date*** instances that are referenced by a dataset.

COLLECTION ITEM

Date: an instance of ***Date*** which is referenced by a dataset.

DatePart

DEFINITION

This class represents a component of a date in a particular calendar system.

DATA MEMBERS

CalendarRef: a pointer to a ***Calendar*** instance, which indicates which calendar system this date was recorded under.

DatePartTypeRef: a pointer to a ***DatePartType*** instance, which specifies the component type

Value: a ***String***, which contains the value of the component. An empty string indicates the value is not known.

EXAMPLE

Data Member	Value
CalendarRef	Pointer to the ***Calendar*** instance with *CalendarID*=1
DatePartTypeRef	Pointer to the ***DatePartType*** instance with *DataPartTypeID*=3
Value	"June"

DateParts

DEFINITION

This collection contains all of the ***DatePart*** instances that are referenced by a ***Date***. A ***Date*** instance owns this collection.

COLLECTION ITEM

DatePart: an instance of ***DatePart*** which is referenced by a ***Date***.

DatePartType

DEFINITION

This class represents the type of a component of a date in a particular calendar system.

DATA MEMBERS

DatePartTypeID: a ***UID*** that uniquely identifies this instance

DatePartTypeName: a ***String***, which is the component name, e.g. Month, Day, or Year.

CalendarRef: a pointer to a ***Calendar*** instance, which indicates which calendar system is using this ***DatePartType***.

EXAMPLE

Data Member	Value
DatePartTypeID	3
DatePartTypeName	"Month"
CelendarRef	Pointer to the ***Calendar*** instance with *CalendarID*=1

DatePartTypes

DEFINITION

This collection contains all of the ***DatePartType*** instances that are defined by a ***Calendar***.

COLLECTION ITEM

DatePartType: an instance of ***DatePartType*** which is defined by a ***Calendar***.

Calendar

DEFINITION

This class represents the calendar system, an instance of ***Date***, was recorded under.

DATA MEMBERS

CalendarID: a ***UID*** that uniquely identifies this instance

Name: a ***String***, which identifies the calendar system, such as "Gregorian", "Julian", etc..

EXAMPLE

Data Member	Value
CalendarID	1
Name	"Gregorian"

Calendars

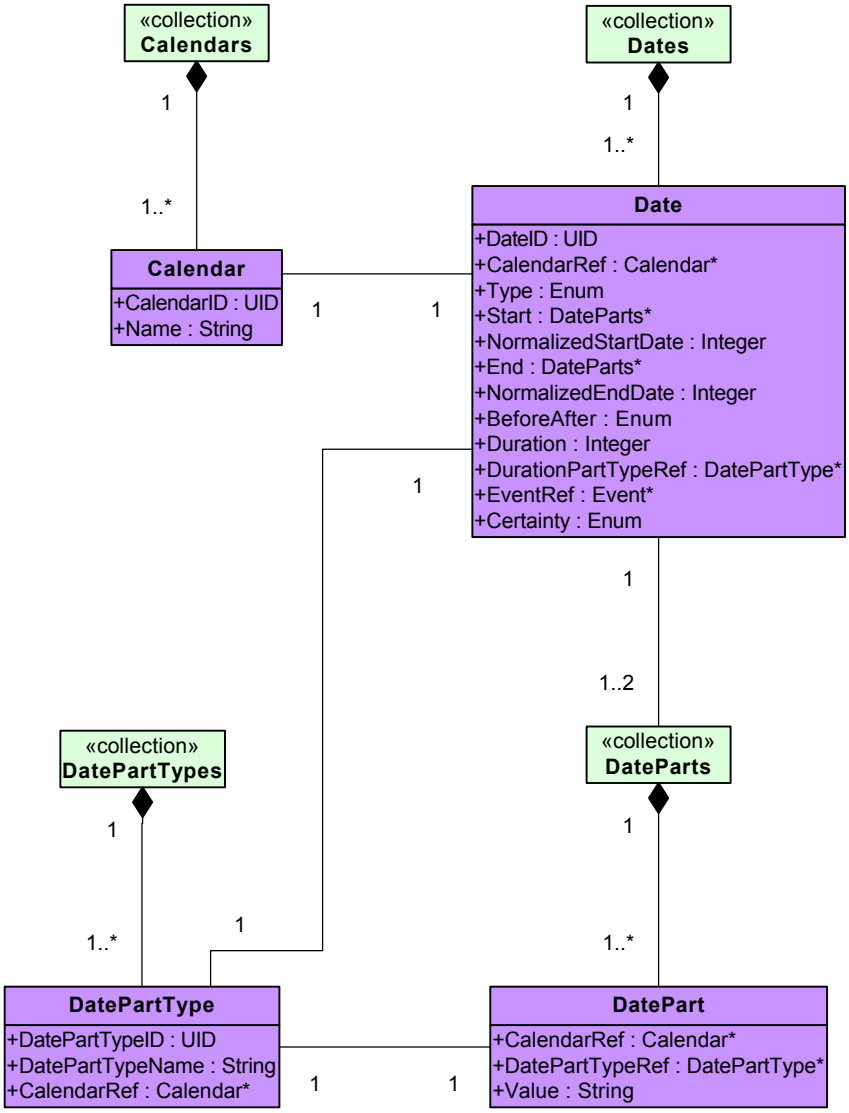
DEFINITION

This class represents the collection of calendar systems which are used by the dataset.

COLLECTION ITEM

Calendar: an instance of ***Calendar*** which is used by the dataset..

Class Diagram:
Date Sub-model



References

1. Chen, P.P. The Entity relational model - Towards a Unified View of Data. ACM Transaction on Database Systems. Vol. 1, No 1, 1976, pp 9-36.

2. Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley: 1999.

3. GENTECH. *Genealogical Data Model, Phase 1. A Comprehensive Data Model for Genealogical Research and Analysis*. May 29, 2000.

4. Dershowitz, Nachum and Reingold, Edward M. 1997. *Calendrical Calculations*. Cambridge University Press, Cambridge, United Kingdom.