**Computationally Independent Model and Service Specification**

**CS13: Specimen Identifier Management Service**

**Version 1.0.3**

|  |  |
| --- | --- |
| **caTissue Team** | Carlos Perez, Poornima Govindrao |
| **Editor** | caTissue Team |
| **Authors** | caTissue Team |

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version** | **Date** | **Author** | **Changes** |
| 0.0.1 | 8/31/2010 | Poornima Govindrao | Initial Document |
| 0.0.2 | 9/1/2010 | Carlos Perez | Updates |
| 0.0.3 | 9/2/2010 | Carlos Perez | Updated to Align with NES Global Unique Identifier service (IS03). |
| 0.0.4 | 9/3/2010 | Carlos Perez | GUID Service in Story Board. |
| 0.0.5 | 9/6/2010 | Carlos Perez | Updated based on review on 9/3. |
| 0.0.6 | 9/9/2010 | Carlos Perez | Update to include resolved issues. |
| 0.0.7 | 9/17/2010 | Carlos Perez | caBIG User Conference updates. |
| 0.0.8 | 9/20/2010 | Carlos Perez | Added more details in to the specification like list of metadata, versioning, error management. |
| 0.0.9 | 9/22/2010 | Carlos Perez | Updated based on Rakesh comments. |
| 0.0.10 | 9/22/2010 | Carlos Perez | Updated based on Ian Fore comments. |
| 1.0 | 11/10/2010 | Carlos Perez | Updated after F2F meeting in Houston |
| 1.0.1 | 12/01/2010 | Carlos Perez | Added registerSite |
| 1.0.2 | 12/06/2010 | Carlos Perez | Update based on Denis Krylov comments |
| 1.0.3 | 12/07/2010 | Carlos Perez | Added entry on transposition errors. |
| 1.0.4 | 12/17/2010 | Carlos Perez | Formatting Update |
| 1.0.5 | 12/20/2010 | Carlos Perez | Formatting Update by Jithu Ramesh |

**Table of Contents**

[1 Overview and Business Case 4](#_Toc316998493)

[1.1 Service Description and Purpose 4](#_Toc316998494)

[1.2 Issues 5](#_Toc316998495)

[1.3 Scope 7](#_Toc316998496)

[1.4 Assumptions 8](#_Toc316998497)

[2 Business Storyboards 8](#_Toc316998498)

[2.1 Primary Actors 8](#_Toc316998499)

[2.1.1 People Actors 8](#_Toc316998500)

[2.1.2 Service Actors 9](#_Toc316998501)

[2.2 Story Boards 9](#_Toc316998502)

[2.2.1 SIDM-SB2 – Fetch GSIDs 9](#_Toc316998503)

[2.2.2 SIDM-SB3 – Fetch parent and/or children information of a specimen 9](#_Toc316998504)

[2.2.3 SIDM-SB4 – Register a specimen with its GSID and parent GSID 10](#_Toc316998505)

[2.2.4 SIDM-SB5 – Register Locally Generated GSID 10](#_Toc316998506)

[2.2.5 SIDM-SB7 – Register Client Site 10](#_Toc316998507)

[3 Detailed Functional Model 10](#_Toc316998508)

[3.1 Structure of the Service 10](#_Toc316998509)

[3.2 Detail of the Capabilities 10](#_Toc316998510)

[3.2.1 Information Model 15](#_Toc316998511)

[4 Profiles 16](#_Toc316998512)

[4.1 Functional Profiles 16](#_Toc316998513)

[4.2 Semantic Profiles 16](#_Toc316998514)

[5 Conformance and Compliance 17](#_Toc316998515)

[5.1 Compliance and Conformance Statements 17](#_Toc316998516)

[6 Appendix A – Relevant Standards 17](#_Toc316998517)

[7 Appendix B - References 19](#_Toc316998518)

[8 Appendix C – Glossary 20](#_Toc316998519)

[9 Appendix D- Dublin Core Meta Data 20](#_Toc316998520)

[10 Appendix E – Open Provenance Model 21](#_Toc316998521)

# Overview and Business Case

This service will manage relationships between specimens across multiple sites/applications to facilitate interoperability. In general, one should be able to get further information about some biospecimen from some distributed set of information services by resolving an identifier that is globally unique. A globally unique identifier is required in order to track a biospecimen’s information as it is processed by different information services.

Biospecimens are commonly shipped from one location to another. The receiving location must be able to electronically acquire relevant biospecimen data (class, type, pathology status, clinical annotations) from the shipping or any other site having information about received biospecimens or their parent biospecimens.

As researchers query for relevant biospecimens globally, biospecimens and their parent and children biospecimens must be associated across one or more biospecimen informatics systems and their annotations must be readily available from all of these systems in order to determine which biospecimens are most relevant to the investigator.

Microarray, proteomics, and next generation sequencing cores are processing and analyzing biospecimens using genome- and proteome-wide profiling technologies. Each of these cores utilizes one or more laboratory information management systems (LIMS) and data delivery systems (e.g. caArray). In the course of experiments performed by these cores, it is also likely that biospecimens will be pooled and combined. These systems must be able to track the provenance of biospecimens which are received, processed, and analyzed.

## Service Description and Purpose

In order to satisfy the above business case, a global Specimen Identifier service should be implemented and hosted centrally per enterprise. This service should be capable of supplying any biospecimen management service with one or more unique identifiers. It should also enable global resolution of identifiers, thus facilitating data agglomeration across relevant Sites containing information about that biospecimen, its parent, or its children (derivatives and/or aliquots).

It is important to emphasize that this service provides functionality that is distinct from a Specimen Management System. A Specimen Management provides the functionality for specimen management from both the clinical and research perspectives. A Specimen Management Service provides support by biospecimen resource facilities providing biospecimen inventory, tracking, and basic annotation.

The Specimen Identifier Management Service by contrast manages the relationships between specimens and their related entities across multiple sites/applications. It may be shared by multiple Specimen Management Systems to ensure interoperability of Specimen Identifiers interchanged by these systems.

For clarity, a Global Specimen Identifer (GSID) is defined as the entity that contains a Globally Unique Idenitifer (UUID) and associated attributes that consist of a name value pair. A UUID is defined as a 16 byte identifier that is intended to be globally unique. The mechanism for its generation is defined in further detail in the document.

## Issues

|  |  |  |
| --- | --- | --- |
| **Short Name** | **Description** | **Resolution** |
| **ID Reservations** | Should there be a two step process for allocating IDs? That is, a reservation is made and then followed by an assignment is made. | ID reservation is not supported by this service. This service as a convenience generated globally unique identifier; however they are not reserved/allocated until an external server explicitly registers these IDs. |
| **User Capabilities** | Business rules may need to be defined for each kind of user that is identified. For example, it may be required that the creator of an ID is the only user that is allowed to edit meta-data of an ID. | In general any operation that changes information in the system requires user credentials. |
| **Attribute Definition** | Are the definitions of the attributes open ended, fixed or defined by a vocabulary server? | Only a limited fixed set of attributes are defined for this service. |
| **GSID Attribute Creation** | How are GSID attributes created? The Specimen Management service may need to specify the attributes that needs to be created or the Specimen Identifier Service may determine what it needs based on querying the Specimen Management service. | Only a limited set of attributes are defined for this service. It is the Specimen Management Service responsibility to provide a more comprehensive set of information. |
| **Query Specification** | What is the format of the query specification? QBE, CQL, SPARQL etc.? | A simple query interface will be employed so that attributes can be queried. The query is not mean to replace a federated query across Specimen Management Services. |
| **Parent/Child Relationships as Attributes** | Is the parent child relationship the same as Attributes? | Convenience operations will exist for the query for all children and parents of an ID. The parent/child relationship shall be maintained as the attribute “parent” of an ID. |
| **Alignment with NES** | Should this service implement functionality that is not present in other NES services particularly the GUID service? For example, the only capability not covered by the GUID service is the query and the ID relation functionality. | The service implements extension to the NES GUID service such as the maintenance of parent/child relationships. The service is compatible with the NES GUID service. |
| **ID Readability** | Is the responsibility of this service to ensure the readability of the generated IDs? For example, should it provide functionality to create human readable IDs? | This is out of scope. |
| **ID Hierarchy and Ordering** | If a client requires IDs that have semantic meaning such as ensuring a sequence of IDs. Should this service provide that functionality? If so, should it employ attributes to support this? | This is out of scope. |
| **ID Aliases** | For local ID that a client would like to maintain. Should this service provide mechanisms to support the maintenance of aliases? | This is not provided by the service. |
| **GSID attribute multi-values** | Should attributes be single or multi-valued? | Attributes are multi-valued. |
| **GSID Immutability** | Can the attributes of a GSID be updated after its initial creation? | All attributes are modifiable. Versioning of updates will be addressed in a future revision. |
| **OID** | Can GSID be represented as OIDs? | See: <http://www.oid-info.com/get/2.25> . For example a UUID of the form: c4e44f20-c595-11df-99a4-0002a5d5c51b can be encoded in an OID of the following form: 2.25.261714135764398102013726101106635162907 |
| **UUID local generation** | Can UUIDs be generated locally? | Yes. UUIDs can be generated locally for subsequent registration. The UUID generation however should derive its value from the MD5 or SHA-1 hash value of the ‘location’ attribute value of the GSID. |
| **Versioning** | If the GSID is updateable, how then can we handle versioning? | Versioning will be addressed in a future revision. |
| **Rights Management** | Should the service support the control of access rights on an individual GSID basis? | Yes. |
| **Tamper Prevention** | Should a watermark or fingerprint be provided to validate a GSID? | This is out of scope. |
| **Transpostion Errors** | Should check digits be embedded in the identifier to check for errors. | A validation check that is performed against an internal database (rather than via an algorithm). |

## Scope

Any service that manages biospecimens can be a client to this service to manage bioidentifiers across multiple services. The use cases identified for this service are as follows:

|  |  |  |
| --- | --- | --- |
| **Items** | **Scope / Out of Scope** | **Source** |
| Providing the ability to register a new Site which will be the consumer of the GSID service. | Scope |  |
| Providing ability to fetch a set of unique IDs on request. | Scope |  |
| Providing ability to fetch parent GSIDs for a specimen with a given GSID | Scope |  |
| Providing ability to fetch children GSIDs for a specimen with a given GSID | Scope |  |
| Providing ability to register a specimen with its GSID and parent GSID(s). | Scope |  |
| Providing ability to fetch information about where the data for a specimen with given GSID resides. | Scope |  |
| Providing ability to retrieve additional information of specimen namely participant demographics, specimen characteristics(tissue Site, pathological status, tissue type) given a GSID | Out of scope |  |
| Participant identification | Out of Scope |  |
| Marking expired Biospecimens | Out of Scope |  |

## Assumptions

|  |  |  |
| --- | --- | --- |
| **Assumption** | **Affects** | **Source** |
| Identifiers generated by this service are assumed to be globally unique. This also implies that when a global unique identifier is created and allocated, it can never be created and allocated again. Stated in another way, a global unique identifier can be assigned to only one Biospecimen. | The Global GSID that is generated must be globally unique. | This Document |
| No semantic meaning should be derivable from the identifier. For example the ordering in time of two identifiers cannot be derived. Another example, the kind of bio-specimen cannot be derived from the value of the identifier. This assumption ensures that a GSID is decoupled from the implementation of the GSID. | The Global GSID should not have any semantic meaning. | This Document |
| Any Site should be able to generate a GSID without being connected to the service. | The Global GSID service is a decentralized service. | This Document |
| The service should be designed to assume an inexhaustible collection of identifiers. | There should be no scarcity of Global GSIDs. | This Document |
| There is no concept of a “use-by-date” for a Global GSID. Once assigned, it is assumed that the identifier will always be used to locate the Biospecimen that it was originally assigned to. | Global GSIDs never expire. | This Document |
| A Global GSID does not change if the attributes associated with the specimen changes. | Global GSIDs are immutable. | This Document |
| The representation of GSID can be presented in many forms. It may be represented by digits, hexadecimal digits, alphanumerics, bar codes, 2d bar codes etc. The minimum requirement is that there exists a mechanism to transform a representation so that it can be resolved into a GSID. | A Global GSID can be represented in many ways. | Out of Scope |

# Business Storyboards

## Primary Actors

The following actors are used in the storyboards below.

### People Actors

|  |  |
| --- | --- |
| **Name** | **Notes** |
| Bench Scientist | Scientist who is interested in getting all available lifecycle information about a specimen regardless of where that biospecimen or its parents/children reside. |
| Administrator | Administrator who will process registration requests to GSID service. |

### Service Actors

|  |  |
| --- | --- |
| **Name** | **Notes** |
| Specimen Identifier Management (GSID) service | This service. |
| Client Site | Any biospecimen management service which will act as a client to this service. |
| Global Unique Identifier (GUID) Service | This service provides capability to uniquely identify all objects (e.g. Persons, Organizations, Studies) as well as their sources. |
| Specimen Management Service (SMS) | A service that manages biospecimens. |

## Story Boards

### SIDM-SB2 – Fetch GSIDs

|  |  |
| --- | --- |
| **Outline** | A biospecimen management Site needs set of GSIDs to be assigned to new specimens being accessioned. |
| **Detail** | * New set of specimens is being accessioned in a biospecimen management informatics Site. * The Site wants to uniquely identify these specimens and would request a centralized service to provide the unique IDs. * In order to request and receive GSIDs the Site must first have requested to be registered to the GSID service. * Administrator reviews and approves registration requests. |

### SIDM-SB3 – Fetch parent and/or children information of a specimen

|  |  |
| --- | --- |
| **Outline** | Fetch parent and/or children information for a specimen |
| **Detail** | A bench scientist utilizing a specimen wants to determine its parent biospecimen and/or its children biospecimen information.  S/he queries the GSID service to retrieve the set of parent and/or children GSIDs and information on where data about each of the specimens reside. This would help the scientist to contact the Site to get more information (e.g. participant demographics, core biospecimen data and annotations). |

### SIDM-SB4 – Register a specimen with its GSID and parent GSID

|  |  |
| --- | --- |
| **Outline** | Register a specimen with its GSID and parent GSID |
| **Detail** | Biospecimen management Site registers the specimens with its GSID and parent GSID(s). |

### SIDM-SB5 – Register Locally Generated GSID

|  |  |
| --- | --- |
| **Outline** | Register Locally Generated GSID |
| **Detail** | A biospecimen management site registers new GSIDs that it generated locally. The biospecimen may have generated this GSID when it was disconnected from the network. |

### SIDM-SB7 – Register Client Site

|  |  |
| --- | --- |
| **Outline** | Register a Client Site |
| **Detail** | A biospecimen information system registers with the GSID. |

# Detailed Functional Model

## Structure of the Service

The following table lists the functionality exposed by the service. Please note that functionality to support the management of this service (i.e. registration of users, monitoring, deployment etc.) is not listed here.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Generate List of UUIDs. | Convenience function to generate UUIDs based on the defined UUID generation algorithm. |
| Register GSIDs | Allows a Site to register a list of GSIDs. |
| Update GSID | Update the attributes associated with a GSID. |
| Retrieve GSID | Retrieve all attributes associated with an input GSID. |
| Retrieve parents of a GSID. | Retrieve all parents of an input GSID. |
| Retrieve children of a GSID. | Retrieve all children of an input GSID. |
| Register Site | Register Site information for the current user. |
| Add Site | Add Site information to an GSID. |

## Detail of the Capabilities

|  |  |
| --- | --- |
| **Name [M]** | Generate List of UUID |
| **Description [M]** | Create a list of new UUIDs. |
| **Pre-Conditions [M]** |  |
| **Security Pre-Conditions [M]** | The client Site has the assigned capabilities to perform this function. |
| **Inputs [M]** | Number of UUIDs to generate. |
| **Outputs [M]** | A List of UUIDs. |
| **Post-Conditions [O]** |  |
| **Exception Conditions [M]** |  |
| **Aspects left for Technical Bindings [O]** | The UUID that is generated is using a hash algorithm. |
| **Notes [O]** | This capability provides a convenience function for generating UUIDs. Another system may generate its own UUIDs if required.  There is no concept of reserving UUIDs or expiring UUIDs. The UUIDs are assumed to be generated uniquely every time this function is invoked.  None of the generated UUIDs are registered into the system. An external system is required to invoke the registration of a UUID as a GSID.  A UUID is distinguished from a GSID in that a UUID is a number that has not been registered with the GSID service. |

|  |  |
| --- | --- |
| **Name [M]** | Register GSIDs. |
| **Description [M]** | Register a list of new GSIDs. |
| **Pre-Conditions [M]** | The GSIDs has not previously been registered. |
| **Security Pre-Conditions [M]** | The client Site has the assigned capabilities to perform this function. |
| **Inputs [M]** | A locally generated list of UUID.  A List of Collections of attributes. The specific list is document at the end of this section. |
| **Outputs [M]** | A list of GSIDs having the same order as the input collection.  If any of the UUIDs inputs are null, a new one is generated. |
| **Post-Conditions [O]** | The attributes are associated to the UUIDs. |
| **Exception Conditions [M]** | Invalid GSID. |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Update GSID |
| **Description [M]** | Updates the associated with a given GSID. |
| **Pre-Conditions [M]** | The GSID has been previously created. |
| **Security Pre-Conditions [M]** | The client Site has the assigned capabilities to perform this function. |
| **Inputs [M]** | GSID and a collection of attributes. |
| **Outputs [M]** |  |
| **Post-Conditions [O]** | The attributes are associated with the input GSID.  Attributes that was not in the collection of the input attributes are not modified.  The attributes that are updated are scoped by the client. |
| **Exception Conditions [M]** |  |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Retrieve GSID |
| **Description [M]** | A client site retrieves attributes associated with a given GSID |
| **Pre-Conditions [M]** | The GSID has been previously created. |
| **Security Pre-Conditions [M]** |  |
| **Inputs [M]** | GSID |
| **Outputs [M]** | The UUID, the relationship with other GSID and an associated collection of collection attributes. Each collection is associated with each client site. |
| **Post-Conditions [O]** |  |
| **Exception Conditions [M]** |  |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Retrieve parents of a GSID. |
| **Description [M]** | Retrieve all parents of a GSID. |
| **Pre-Conditions [M]** | The GSID has been previously created. |
| **Security Pre-Conditions [M]** |  |
| **Inputs [M]** | GSID |
| **Outputs [M]** | A graph that begins with the root and contains only leafs that are parents of this GSID. |
| **Post-Conditions [O]** |  |
| **Exception Conditions [M]** | GSID does not exist. |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Retrieve children of a GSID |
| **Description [M]** | Retrieve all children of a GSID. |
| **Pre-Conditions [M]** | GSID does not exist. |
| **Security Pre-Conditions [M]** |  |
| **Inputs [M]** | GSID |
| **Outputs [M]** | A graph that has this GSID as its root and all children of this GSID as its siblings. |
| **Post-Conditions [O]** |  |
| **Exception Conditions [M]** |  |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Add Site |
| **Description [M]** | Add Site information for a GSID |
| **Pre-Conditions [M]** | The user is logged into the system. |
| **Security Pre-Conditions [M]** |  |
| **Inputs [M]** | •GSID |
| **Outputs [M]** | Ok |
| **Post-Conditions [O]** | The Site that has been registered by the current user is associated with the GSID. |
| **Exception Conditions [M]** | GSID does not exist. |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

|  |  |
| --- | --- |
| **Name [M]** | Register Site |
| **Description [M]** | Register Site information for the current user. |
| **Pre-Conditions [M]** | The user is logged into the system. |
| **Security Pre-Conditions [M]** |  |
| **Inputs [M]** | • application  • application URL  • application version  • contact email  • contact name  • contact phone  • organization |
| **Outputs [M]** | Ok |
| **Post-Conditions [O]** |  |
| **Exception Conditions [M]** |  |
| **Aspects left for Technical Bindings [O]** |  |
| **Notes [O]** |  |

### Information Model



Figure 2.



Figure 3.

# Profiles

## Functional Profiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Profile No.** | **Functional Profile Name** | **Functional Profile Description** | **Capability Name** |
| SIDM-FP1 | GSID Query | Read only access | * Retrieve parents of a GSID. * Retrieve children of a GSID. * Retrieve GSID * Generate List of UUID |
| SIDM-FP2 | GSID Edit | Registered Sites | * Create New GSID * Update GSID * Add Site * Register Site |

## Semantic Profiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Semantic Profile No.** | **Semantic Profile Name** | **Constrained Information Model** | **Semantic Profile Description** |
| SIDM-SP1 | GSID | BRIDG 2.0 | GSID Service |

# Conformance and Compliance

## Compliance and Conformance Statements

| **Name** | **Type** | **Viewpoint** | **Description** | **Test method** |
| --- | --- | --- | --- | --- |
| Query Performance | Obligation | Engineering | The GSID service should provide a response within 0.5 seconds to support a synchronous UI based client | Test cases to include performance testing. |
| Multiple Jurisdictions | Obligation | Enterprise | The GSID service will span jurisdictional boundaries and will need to support a federated data model. | Test cases include multiple domain scenarios. |
| Additional Functionality | Permission | Computational | The GSID service can provide additional functionality other than specified in these specifications | Design Review |
| Data Types | Obligation | Informational | The GSID service must conform to NCI’s constrained list of ISO 21090 data types. | Design Review |
| Functional Profiles | Obligation | Computational | Functional Profiles shall be deployed as functional wholes. Ignoring or omitting functional behavior defined within a functional profile is not permitted, nor is diverging from the detailed functional specifications provided in Section 4. | 1. Design Review 2. Test cases |
| Functional Profiles – Conformant Implementation | Obligation | Computational | A conformant implementation of this specification must deploy at least one Functional Profile. If that Functional Profile has dependencies on other profiles, then those dependencies must be deployed as well to support the provenance of that service Site. | 1. Design Review 2. Test cases |

# Appendix A – Relevant Standards

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Location** |
| Open Provenance Model | The Open Provenance Model is a model of provenance that is designed to meet the following requirements: (1) To allow provenance information to be exchanged between systems, by means of a compatibility layer based on a shared provenance model. (2) To allow developers to build and share tools that operate on such a provenance model.(3) To define provenance in a precise, technology agnostic manner. (4) To support a digital representation of provenance for any \thing", whether produced by computer systems or not. (5) To allow multiple levels of description to coexist. (6) To define a core set of rules that identify the valid inferences that can be made on provenance representation. | <http://openprovenance.org/> |
| HL7v3 | Health Level 7 version 3 | <http://www.hl7.org/implement/standards/v3messages.cfm> |
| Handle Service | The Handle Service is an infrastructure on which applications serving many different purposes are being built. Some examples are rights management applications, persistent identifiers for digital objects on the Web, and institutional data preservation and archiving. | <http://www.handle.net/> |
| Handle.net Globus Integration | The goal of this "Handle System – Globus Toolkit Integration Project" is to leverage CNRI's Handle System technology through GT's Web services' protocols. | <http://www-unix.globus.org/toolkit/projects/handle_system.html> |
| DOI | The Digital Object Identifier (DOI®) Service is for identifying content objects in the digital environment. DOI® names are assigned to any entity for use on digital networks. They are used to provide current information, including where they (or information about them) can be found on the Internet. Information about a digital object may change over time, including where to find it, but its DOI name will not change. | <http://www.doi.org/> |
| PURL | PURLs (**P**ersistent **U**niform **R**esource **L**ocators) are Web addresses that act as permanent identifiers in the face of a dynamic and changing Web infrastructure. Instead of resolving directly to Web resources, PURLs provide a level of indirection that allows the underlying Web addresses of resources to change over time without negatively affecting services that depend on them. This capability provides continuity of references to network resources that may migrate from machine to machine for business, social or technical reasons. | <http://www.purl.net/docs/index.html> |
| PURL Federation Architecture |  | <http://code.google.com/p/persistenturls/wiki/PURLFederationArchitecture> |
| ISO UUID | An OID is a tree structured series of numbers separated with '.' (dot) and is read from left to right. | <http://www.itu.int/ITU-T/asn1/uuid.html> |
| ISO X.660 | OID Naming standard | <http://www.oid-info.com/standards.htm> |
| Candidate NCI Enterprise Services | Current Listing of candidate NCI Enterprise Services | <https://wiki.nci.nih.gov/display/EAWiki/Candidate+NCI+Enterprise+Services> |
| NES Global Unique Identifier Service | caGrid Identifiers Framework CIM | <https://ncisvn.nci.nih.gov/WebSVN/listing.php?repname=cagrid&path=/branches/caGrid-1_4_release/Documentation/core/Identifiers/ECCF/> |
| Dublin Core | The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description. | <http://dublincore.org/documents/dces/> |
| UUID | UUID Wikipedia Entry | <http://en.wikipedia.org/wiki/Universally_unique_identifier> |
| MID | Machine Identifier | <http://wiki.freebase.com/wiki/Mid> |
| Tinify | Freebase GUID Compactor | <http://tinyify.freebaseapps.com/> |
| Provenance Vocabulary Mappings | Identifies correspondence between a set of core provenance concepts defined in the Open Provenance Model (OPM) and other provenance terminologies. | <http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings> |

# Appendix B - References

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Location** |
| Open Provenance Model | The Open Provenance Model is a model of provenance that is designed to meet the following requirements: (1) To allow provenance information to be exchanged between systems, by means of a compatibility layer based on a shared provenance model. (2) To allow developers to build and share tools that operates on such a provenance model. (3) To define provenance in a precise, technology agnostic manner. (4) To support a digital representation of provenance for any “thing", whether produced by computer systems or not. (5) To allow multiple levels of description to coexist. (6) To define a core set of rules that identify the valid inferences that can be made on provenance representation. | <http://openprovenance.org/> |
| FU | HTTP Versioning Mechanism for LinkedData | <http://events.linkeddata.org/ldow2010/papers/ldow2010_paper13.pdf> |

# Appendix C – Glossary

|  |  |
| --- | --- |
| **Term** | **Description** |
| BRIDG | The Biomedical Research Integrated Domain Group (BRIDG) has developed a comprehensive domain analysis model representing biomedical/clinical research. |
| LSDAM | The caBIG Life Sciences Domain Analysis Model (LSDAM) is a shared view of the semantics of the Life Sciences domains: Integrative Cancer Research (ICR), Tissue Banking and Pathology Tools (TBPT), and Imaging, and is aligned, where appropriate, with the Clinical Trials Management Systems (CTMS) workspace (i.e., BRIDG). It is intended to provide the foundation for semantic interoperability among the various applications within the caBIG Life Sciences domain. |

# Appendix D- Dublin Core Meta Data

Example mapping of Dublin Core to FGDC Biological Data Profile

|  |  |  |
| --- | --- | --- |
| **Dublin Core** | **Description** | **Bio Profile** |
| title | A name given to the resource. | Title |
| date | A point or period of time associated with an event in the lifecycle of the resource. | Single Date, Ending Data, Last date entered in a multiple data range. |
| subject | The topic of the resource. |  |
| type | The nature or genre of the resource. |  |
| date.created | A point or period of time associated with an event in the lifecycle of the resource. | Metadata Date Created |
| date.modified | A point or period of time associated with an event in the lifecycle of the resource. | Metadata Review Date |
| temporal DCMI period |  | Age Estimate |
| creator | An entity primarily responsible for making the resource. | Originator |
| description | An account of the resource. | Abstract |
| publisher | An entity responsible for making the resource available. | Primary Contact Organization |
| contributor | An entity responsible for making contributions to the resource. | Data Set Credit |
| coverage | The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant. |  |
| identifier | An unambiguous reference to the resource within a given context. | Online Linkage |
| source | A related resource from which the described resource is derived. | Lineage |
| format |  | Non-digital form (or) Format Name |
| rights access | Information about rights held in and over the resource. | Access Constraints |
| rights use |  | Use Constraints |

One possible exercise is to map BioSpecimen information into the dublincore metadata elements.

# Appendix E – Open Provenance Model

The Open Provenance Model (OPM) goal is to capture the causal dependencies between artifacts, processes, and agents. A provenance graph is defined as a directed graph, whose nodes are artifacts, processes and agents, and edges that represent a causal dependency between its source and its destination. The categories of edges are depicted in the figure:



***Figure 4. Categories of edges in the Open Provenance Model (OPM).***

The circles represent artifacts, the squares represent processes and the octagons represent agents. Also note that for OPM edges “used" and “wasGeneratedBy" can be extended with an optional timestamp, indicating that the associated artifact was known to be generated or used, at a given time.

If we were to treat biospecimens as artifiacts, then only the “wasDerivedFrom” edge is explicitly covered by the specication. This is because processes and agents are not first class entities in this specification. Although one may be able to represent “wasGeneratedBy” edge may as metadata associated with the GSID. The other edges (i.e. used, wasControlledBy, wasTriggeredBy) are not captured in the semantics of this service.

The OPM prescribes the following metadata associated with its entries:

|  |  |
| --- | --- |
| **type** | Subject: an annotable entity property: <http://openprovenance.org/property#type> value: a URI meaning: Denotes the subtype of an OPM entity. Such sub-types are represented by a URI. |
| **pname** | Subject: an annotable entity property: <http://openprovenance.org/property#pname> value: a URI meaning: Denotes a persistent name that can be used by OPM graph queriers to compare OPM entities. The scope of this name is intended to be global. |
| **label** | Subject: an annotable entity property: <http://openprovenance.org/property#label> value: a String meaning: This property provides a human-readable version of an OPM entity. |
| **value** | Subject an artifact property: <http://openprovenance.org/property#value> value: a typed value meaning: Denotes a serialization of an application value associated with an OPM entity. Such serialization should have a type (expressed in a type system suitable for the serialization). Serialization technologies include XML, JSON, and ntriples. |
| **encoding** | Subject: an artifact or an OPM graph property: <http://openprovenance.org/property#encoding> value: a URI meaning: Denotes how a serialization was constructed. For instance, using the Java bean serialiser to create anXML document, by applying a specified transformation to the application data, e.g. anonymisation, by passing a reference to the actual value, or by creating a set of RDF triples. |
| **profile** | Subject: an OPM graph property: <http://openprovenance.org/property#profile> value: a URI meaning: This property applies to an OPM graph and denotes a profile that is supported by that graph. |