Effect of SAIF-ECCF Adoption on Common Data Element Reviews

Mukesh Sharma[1], Lynne Wilkens[2], Claire Wolfe[3], Pankaj Agarwal[4], Xin Zheng[5], Dianne Reeves[6], Brian Davis[7]

[1]Washington University in St. Louis; [2]Cancer Research Center of Hawaii, University of Hawaii; [3]TerpSys; [4]Duke University; [5]Albert Einstein College of Medicine; [6]National Cancer Institute-Center for Biomedical Informatics and Information Technology; [7]3rd Millennium, Inc

June, 2010

Table of Contents

1 Introduction 4

2 History and Background 4

2.1 Current caBIG® Interoperability Use Case 4

2.2 What is a Common Data Element? 5

2.2.1 Definition 5

2.2.2 Operationization of interoperability via CDEs in caBIG® 5

2.2.3 Review of CDEs 6

2.3 Statement of problem 6

2.3.1 Issue of reuse of CDE 6

2.3.2 ISO 21090 datatypes 7

2.3.3 Adoption of SAIF/ECCF environment 7

3 Future 7

3.1 Future caBIG® Interoperability Use Case 7

3.2 Working Interoperability 8

3.3 SAIF/ECCF environment 9

3.3.1 SAIF 9

3.3.2 ECCF 11

3.4 Where do semantics fit in ECCF 13

3.4.1 CDEs and ECCF 13

4 Recommendations 14

4.1 Recommendations for Semantic Requirements 14

4.1.1 Computationally Independent Model (CIM) 15

4.1.2 Platform-Independent Model (PIM) 16

4.1.3 Platform-Specific Model (PSM) 17

4.2 Recommendations for Reviews 19

4.2.1 Current Reviews 19

4.2.2 Semantic Reviews in context of ECCF 19

4.2.3 Roles of the VCDE in review of Semantics in ECCF 20

4.3 Additional Recommendation 21

5 Conclusion 21

6 References 22

7 Appendix 23
7.1 Metadata in caDSR 23
7.1.1 Metadata Components 23
7.1.2 Required Data Element ISO 11179 attributes in caDSR 24
7.2 Specifics on how interoperability is currently achieved on the caGrid 26
7.2.1 Key technology enablers which allow semantic and syntactic interoperability 26
7.2.2 Semantic Interoperability 26
7.2.3 Syntactic interoperability 27

1 Introduction

The goal of this white paper is to discuss the effect of the adoption by cancer Biomedical Informatics Grid® (caBIG®) of the Enterprise Conformance and Compliance Framework (ECCF)-Services-Aware Enterprise Framework (SAIF) on Common Data Element (CDE) Reviews that take place within the Vocabulary and Common Data Elements Workspace (VCDE WS). In particular, the paper will discuss the use and review of data elements and their components in the present caBIG® environment and how these elements should be considered in the SAIF/ECCF environment.

2 History and Background

2.1 Current caBIG® Interoperability Use Case

The goal of caBIG® is to enable data integration between heterogeneous data sources in order to enable implementation of complex scientific use cases. Data integration enables identification of implicit relationships between various sources of data, which in turn allows new knowledge discovery. The first step toward data integration is achieved through interoperability, the ability of information systems to both access and appropriately use data from a remote data resource. Syntactic interoperability enables systems to exchange data through shared interfaces. Semantic interoperability occurs when systems are able to "understand" the data exchanged.

The use case that has driven caBIG® interoperability for 5 years has been that an organization/person (either funded by caBIG® or external to caBIG®) has data or a software service to share with many others on the caGrid. caBIG® took a proscriptive approach to achieving interoperability. The idea has been to use interfaces as the central point for integration. Further, the idea was to describe the interface as a UML model using concepts from a controlled vocabulary. Semantic interoperability would be achieved by the use of controlled vocabularies, and syntactic interoperability would be achieved by the rules regarding artifact registration on the caGrid and metadata registration in the cancer Data Standards Registry and Repository (caDSR). To achieve full interoperability, Common Data Elements (CDEs) need to be re-used between applications as this allows the caGrid to aggregate data through SQL queries.

The current process for becoming caBIG® compatible and sharing data or services on the caGrid is as follows. The interface needs to be represented as a UML model, where the objects, attributes and value domains are annotated with a controlled vocabulary that is available in LexEVS hosted at National Cancer Institute Center for Biomedical Informatics and Information Technology (NCI-CBIIT). Furthermore, the annotated UML model must be converted to metadata with the NCI-CBIIT administrated "UML Loader" and stored in the metadata repository known as the caDSR. The Introduce tool is used to register the software service on caGrid and to construct an XML schema that is aligned with the UML model. Further, to achieve semantic and syntactic interoperability on the caGrid, certain important data elements called "COMMON Data Elements or CDEs" need to be re-used, as that will allow the caGrid to aggregate data through SQL queries. The specific details of how interoperability is achieved on the caGrid are given in the Appendix.

2.2 What is a Common Data Element?

Common data elements (CDEs) have been the central aspect of interoperability within caBIG® for the last 5 years. The VCDE WS, the Architecture WS and the caCORE team at NCI-CBIIT have looked to CDEs, and associated tools and processes, to provide "seamless interoperability on the Grid". Below is a brief description of CDEs and their use in caBIG®.

Data Elements are stored in the NCI caDSR or any ISO 11179 compliant metadata repository as metadata. Metadata is defined as data about data; it defines and identifies a unit of data. The ISO 11179 data standard was adopted by caBIG® and the NCI-CBIIT for the formations of CDEs.

2.2.1 Definition

The ISO 11179 Standard defines a data element as a "unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes" (International Standard ISO/IEC 11179-1, second edition 2004-09-15). The caDSR system defines a data element as the "smallest unit of information that can be exchanged in a transaction between cooperating systems" (caDSR, Release 4.0 Technical Guide).

A Common Data Element (CDE) is simply another name for a Data Element as described above. CDE is the name used by the caBIG® community to emphasize the importance of reuse of data elements among the services on the Grid. A Data Element that describes a unit of information in two different services is common to those two services. Common Data Elements identify points of interoperability among services - if two services contain data described by the same CDE, then that data in those two services are identical in terms of meaning (semantics) and representation (syntax).

A Data Element is composed of the ISO 11179 standard components: the Data Element Concept (DEC) and the Value Domain (VD). The DEC unambiguously describes what a Data Element is. The VD describes the representation of the Data Element. CDEs are currently uploaded to the caDSR by the UML Loader tool or manually curated. The current UML Loader requires certain attributes described in the ISO 11179 standard to be specified in order to create components for storage in the caDSR. The major components that make up a Data Element in the ISO 11179 standard and/or in the caDSR are described in the Appendix. Many of the elements required in the caDSR are not needed to create CDEs, but instead describe the UML model, such as Classification Scheme Items (UML_PACKAGE_NAME type).

2.2.2 Operationization of interoperability via CDEs in caBIG®

Different applications within caBIG® interoperate with each other based on the principles of Service Oriented Architecture (SOA). This requires the interaction between the applications to be location independent and based on an interoperable message format. Location independence is achieved by interaction on the caGrid, and interoperable messaging is achieved by XSD (XML Schema Definition) language. In brief, caBIG® data sources and tools are "exposed" on the caGrid through SOAP based APIs that exchange XML messages in a format defined by XSD. Use of XSD tied to CDEs enables a "common" understanding of the contents of the XML messages by the different applications. For two systems to be

semantically interoperable, they must be able to communicate the meaning of each piece of data in their system. If the two systems use the same Data Element to describe these pieces of information, then we know that they are the same. More specifics on interoperability are given in the Appendix.

2.2.3 Review of CDEs

There are presently two types of CDE reviews that the VCDE workspace developed and performed. One of the requirements for an application to become caBIG® sliver level compatible is that it use well-formed and annotated CDEs and reuse CDEs wherever possible. During the silver level review, each CDE is assessed for a clear and consistent definition, as well as reuse of existing CDEs. Full reuse occurs when two services use the same CDE. However, any ISO 11179 component may be reused by multiple systems. Therefore, two systems can semantically interoperate at any of those levels (Data Elements [DEs], Data Element Concepts [DECs], Object Concepts [OCs], Property, Value Domains [VDs], Value Meanings).

CDEs can be promoted to caBIG® standard CDEs by being approved after a review process that shows that the CDEs are viewed as standards by the content area experts. These data elements have is intended to provide maximal reuse across application.

2.3 Statement of problem

Interoperability on the caGrid between data services and applications has been partly successful. There are many issues that require evaluation of our current approach to achieving interoperability.

2.3.1 Issue of reuse of CDE

The linkage of concepts to controlled vocabularies has successfully ensured that data elements are clearly defined. However, the implementation of the ISO 11179 standard in the caDSR has required reuse of data at the CDE level and the configuration of the caGrid has required reuse at the Object Class level.

In reality, full CDEs reuse has been rare in caBIG®, while partial reuse, especially at the concept level for attributes occurs quite often. For example, Patient.address cannot fully reuse the CDE of Person.address, but the concept of address is shared by the two scenarios.

Similarly, full reuse of CDEs requires that the same value domain be used. Therefore, mapping will not occur between birthdates, where one is represented as the month and year, and the other includes month, year, day and time.

Other barriers to interoperability with the current system include the following. The process of loading metadata from a UML model into the caDSR and building a Grid service is cumbersome with a steep learning curve. The current system encourages development under UNIX and java systems, while interoperability should be platform independent. Similarly, the implementation of CDEs based on the ISO 11179 standard in caDSR blends the concepts of "semantic interoperability" and "syntactic interoperability" (for example, classification scheme above is a syntactic interoperability element), while CDEs should be completely platform independent.

Predefined Value Domains that represent "generic" data types like numbers or strings are all mapped to Value Domains with Java specific data types, which are not platform independent. Lastly, there is currently little ability to interoperate during workflow on caGrid. "There is no way to link a registered model to a run-time schema representation, a fact that directly impedes predictable Computable Semantic Interoperability (CSI)."(https://wiki.nci.nih.gov/x/tSdLAQ).

2.3.2 ISO 21090 datatypes

caBIG® has adopted the ISO 21090 standard for data types, which includes complex data types such as code sets (CD) and dates (DT). These datatypes should aid interoperability. For example, dates that include different components will all have datatype DT and be mapped together. While ISO 21090 data types have been loaded into caDSR by modeling the data type as a data element in UML, the current implementation of the ISO 11179 standard in the caDSR does not allow for the full deployment of the ISO 21090 data types, such that all components of a complex data type are bound together and different flavors of the data types (inclusion of a subset of components only) can be accommodated.

2.3.3 Adoption of SAIF/ECCF environment

For a variety of reasons, including some of the ones mentioned here, the caBIG® community is migrating to a SAIF/ECCF environment. The process of handling of metadata and semantics needs to be reconsidered under this new paradigm. In fact, all semantic artifacts from ECCF cannot be handled in the ISO 11179 standard, such as the semantic documents related to semantic profiling. This juncture also provides a good time to reflect on the best practices regarding semantic metadata generally.

3 Future

3.1 Future caBIG® Interoperability Use Case

Two use cases that will drive caBIG® in the future.

In the first use case, an organization/person wishes to cooperate with caBIG® and achieve interoperability, especially around a certain type of data or process or software service. Under the SAIF/ECCF environment, caBIG® will have a Business Architecture Model (BAM), Domain Analysis Model (DAM) and specification stacks of relevant services. These artifacts will be used to identify and define points of interoperability between caBIG® and other organizations. Note that caBIG® has no "proscriptive powers" over these developers. caBIG® cannot force these developers to adopt BAM, DAM, Specifications stack rules, nor use of NCI CBIIT technologies and tools. The process of interoperability will thus begin with a review and discussion of what might be achievable in specific areas on which interoperability is desired.

In the second use case, developers are funded within caBIG®/NCI-CBIIT and need to ensure that they are building applications that will interoperate with other caBIG® and NCI-CBIIT services. The ECCF's Enterprise or Business Viewpoint should have already considered where this service fits into the overall priority scheme of the organization (using the BAM and also perhaps enterprise use cases, or other business use case prioritization mechanisms). These developers need to be familiar with and to either adopt or adapt to:

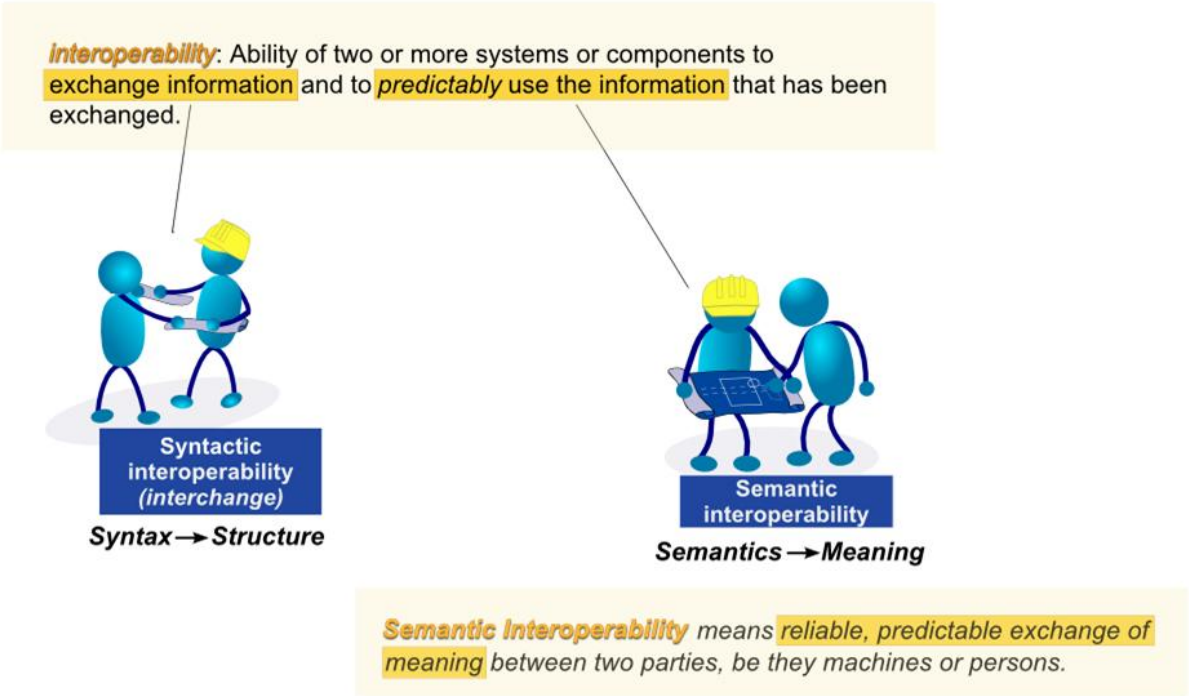
- caBIG® Business Architecture Models (BAM)
- caBIG® Domain Analysis Models (DAMs)
- caBIG® specification stack templates

That is, caBIG® and NCI CBIIT have proscriptive power over these developers.

To address these future use cases caBIG® will depend on working interoperability and the SAIF-ECCF environment

3.2 Working Interoperability

Working Interoperability (WI) is the collection of structures, processes, and components that support Computable Semantic Interoperability (CSI) between two parties ("trading partners") who are interacting (for example, exchanging information, coordinating behavior) to achieve one or more business goals. Interoperability can be further defined to be the deterministic exchange of data (syntax) or information in a manner that preserves shared meaning (semantics)



CSI between two or more trading partners in collaboration/interaction needs:

- Pre-transaction agreement, such as some form of a contract that clearly details the specifics of the expected results of the transaction. This agreement can best described by the four pillars of CSI.
    1. A common domain analysis model of shared semantics (static and dynamic) - Models exist at several levels of abstraction, the most common being at the Analysis or Information model which identified the core classes, attributes, and relationships in a domain. Terminology and data type models are also examples. Most importantly, they must be implementation independent.

2. Computable complex data type specification - All artifacts require shared data type representation such as ISO 21090.
3. A rigorous and reproducible methodology for binding static attributes to terms from concept based terminologies.
4. A formally-defined process for defining the static structures and interaction profiles that collectively define the particular information that is exchanged between trading partners

- Post-transaction assessment of the degree of success of the transaction in progress or immediately following completion.

To define these "WI transactions", a framework that enables WI (e.g. SAIF) must provide a conformance and compliance Framework (e.g. ECCF), and, in alignment of pillars of CSI, a set of constructs that allow for the explicit expression of the static, functional, and behavioral semantics.

Note: This section is extracted from -SAIF Interoperability Reviews -White Paper; background; https://wiki.nci.nih.gov/x/TJuPAQ.

3.3 SAIF/ECCF environment

3.3.1 SAIF

The services aware information framework (SAIF) is a framework for computable semantic interoperability. SAIF is a way of thinking about producing specifications that explicitly describe the governance, information, and behavioral semantics that are needed to achieve computable semantic working interoperability. The intended information transmission technology might use a messaging, document, or services exchange approach. SAIF defines the artifacts and specification semantics needed to support interoperability in healthcare, life sciences, and clinical research. The main Value proposition is "working interoperability" and primarily, service aware. This does not dictate implementation via a service such as SOA. Service awareness informs SAIF from the perspective of separation of concerns (behavioral vs. static semantics) with explicit representation of contracts and interactions.

3.3.1.1 Guiding Principles of SAIF

- SAIF is both a pre-condition for "well-designed" services aware enterprise architecture and a component of an SOA.
- SAIF provides for "working interoperability" which implies collection of structures, processes, and components that support Computable Semantic Interoperability (CSI) in a Conformance and Compliance Framework.
- SAIF facilitates the explicit and layered expression of the set of static, functional, and behavioral semantics that collectively enable Working Interoperability.
- The specifications to enable Working Interoperability are defined in such a manner so as to ensure that they are usable, useful, and durable. These specifications are implementable in a variety of deployment contexts in a repeatable, understandable manner.

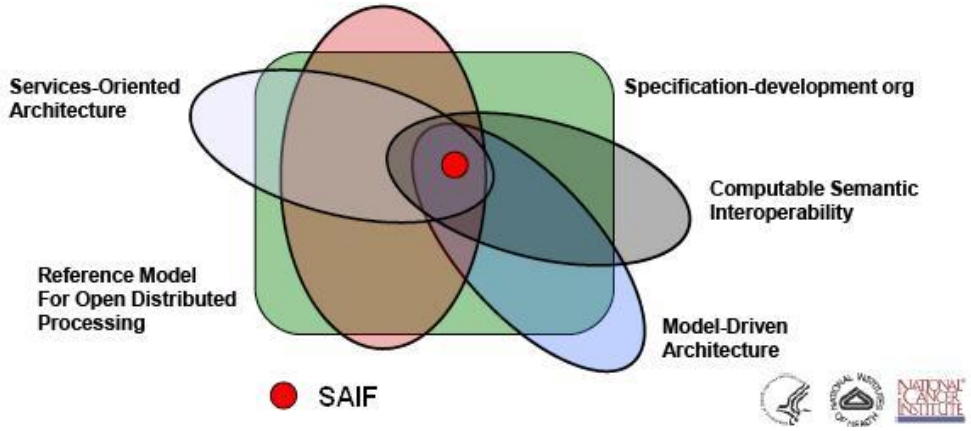
- Trading partners can participate in a specific business interaction without making assumptions about the size, character, or identity of the other partner, and no assumptions are made for the exchange details (what, how, when, etc.).
- Two "trading partners" interoperate based on a certified "level of shared compliance" to interoperability specifications or standards.
- Certified "level of conformance" determines the degree of automated interoperability that is possible and/or the difficulty of the transformations that are required to enable interoperability.

3.3.1.2 Components of SAIF

SAIF defines four frameworks that, together, provide context, processes, and specific deliverables for deriving HL7 enterprise architecture specification. These sub-frameworks provide a "grammar" for describing those aspects of enterprise architecture that impact "working interoperability". These four frameworks include the following:

- The Behavioral Framework - Provides a specification of integration semantics of IT components and linkage of integration semantics to real-world behaviors. This provides for semantics of operations, interactions, and collaborations. The BF formalizes various aspects of interactions through role, contract, and interaction. We see this in the "stairway to heaven" graphic as the CIM, PIM and PSM levels, using the RM-ODP language and this will population the computational viewpoint of the ECCF.
- The Enterprise Conformance and Compliance Framework (ECCF)-The ECCF is the key component of SAIF. The ECCF provides a specification stack (SS) template which can use to collect testable conformance statements. Using conformance assertions, systems can be certified against these conformance statements. The ECCF is a layered approach to enable different degrees of compliance and conformance.
- The Governance framework - based on HL7 governance, deals with relationships between HL7 and other organizations that specify standards, the relationship between the standards and governance, and governance within an interoperability community.
- The Information Framework- this provides static semantics and existing information framework artifacts. Examples of the static semantics include information models, data models, domain analysis models, data type bindings (ISO 21090), deployment topologies (UML component model) and business rules. These will populate the information viewpoint of the ECCF as well as the business/enterprise and engineering viewpoints.

3.3.1.3 Lenses of SAIF



SAIF is an intersection of SOA, Model-Driven Architecture (MDA), CSI, Distributed Systems Architecture, and an organizational context. It provides goals, artifacts, portions of a methodology, and a framework for defining an Enterprise Architecture. It uses robust, durable, business-oriented constructs that provide extensibility, reuse, and governance.

3.3.2 ECCF

The ECCF focuses on providing a framework for developing software specifications that can be implemented and achieve certifiable, scalable, and tractable working interoperability in a number of deployment contexts. Conformance of a technology binding to a given conformance statement is claimed using a conformance assertion. Compliance of an artifact source to target transformation (i.e. defined by a constraint pattern) is claimed by a given specification but not usually formally tested.

A specification Stack is essentially a 3-row X 4 -column matrix where Rows are levels of abstraction that map to Object Management Group's (OMG) Model-Driven Architecture (MDA) and columns map to the Reference Model of Open Distributed Processing (RM-ODP) viewpoints. Each cell of the matrix contains explicit collection of conformance statements for a given area that are sorted by the RM-ODP viewpoint, leveled by MDA levels of abstraction and Validated using technology binding. Cell to Cell or Row to Row navigation is specified by compliance constructs.

Subject Specification	Enterprise / Business Viewpoint	Information Viewpoint	Computational Viewpoint	Engineering Viewpoint
CIM (Conceptual)	Scope and Vision document Business objectives Use Case Inventory Non-functional requirements	Domain Analysis Model	Service inventory Service interface CFSS Service roles and relationships	To be populated as needed
localizations				
PIM (Logical)	Use Case Specification Wireframes Business rules Industry technology-neutral standards	Domain Information Model Constrained Information Model Localized Information Model	Service Interface PIM Collaboration Specification	To be populated as needed
localizations				
PSM (Implementable)	Industry technology-specific standards	Database schema Message schema Transformations	Service Interface PSM Service Realization Specification Orchestration scripts	Application and User Interface component design Deployment Model Topology
localizations				

The value of the ECCF is that prevents implicit assumptions which are the enemy of working interoperability. The explicit assumptions are referred to as conformance statements. Implementations can make pair-wise conformance assertions against conformance statements.

This allows for testable, verifiable and certifiable conformance. It provides a structured way to certify conformance with implementation testability to stated conformance statements and verify compliance with explicit transformations/utilization of the specifications/standard artifacts, and integrate external developed specifications and standards. An organization adopting the ECCF can build explicit specifications and standards that are compliant with other specifications and and standards, and explicitly express business rules and localizations which are the critical enterprise architecture requirements that enable Working interoperability to be achieved.

For caBIG®, ECCF adoption will result in key changes from the past. First, all applications and Tools are developed as services. The requirements, business rules and objectives are tied to a service capability. Testable conformance and compliance statements are provided about the services. The capabilities of the service are organized using functional and semantic profiles. Service dependencies and sequences of interactions are captured and made explicit. The layered specification allows support for different levels of interoperability and traceability. This framework provides a more structured and rigorous specification that captures additional information that allows for better definition of the service. It supports ISO 21090 data types and leverages the DAM's whenever possible.

The implementation of the guidelines for ECCF is at two audiences: The project/service providers are scoped for "how to build a service", providing for the specification of a system as a set of services that can then be re-used by other to implement those same services. This is at the PIM level. The second scope involves users or clients of the services, for specifying "how to use a service" The manner by which these can be adopted is through a set of templates.

3.3.2.1 The OMG MDA levels of abstraction

- Computationally-independent Model (CIM) - This is the conceptual view. This is defined as part of the requirements and analysis phase and maps requirements to a set of functions, capabilities or behaviors. It defines the boundaries and provides traceability to business objectives. It maps to the business objectives, purpose and scope of the service. This organizes the requirements for the service as a set of capabilities (i.e. the system should support "describe capability" and the CIM defines the concepts required to capture the domain of interest (i.e. DAM). It also captures the business constraints and policies.
- Platform Independent Model (PIM) - This is the logical view of the service/system, also developed during design phase. It transforms the analysis model into a logical model without binding to a specific platform. It maps the services in the CIM to a set of services and refines the structures as necessary. Captures platform independent logical constructs like business rules, captures logical dependencies and interactions with other services, and captures the implementation policies and constraints without binding to a platform.
- Platform-specific Model - (PSM) - this is the implementable view. Usually defined during the design and implementation phases. The PSM provides Platform specific models, bindings and mappings with platform specific deployment artifacts. (WSDL, Orchestration, scripts, etc.), and captures platform specific business rules, package and deployment models, and policies and constraints. A given implementation "binds to /Realizes" a given ECCF specification. The conformance statements made at the CIM/PIM/PSM are bound to pair-wise Conformance Assertions in the implementation, thereby allowing for tractable certification process.

3.3.2.2 Data Services under ECCF

A data service is typically a "long-lived" service which can be characterized by having input data and output data (informational point of view) which may be the same, and methods are simple, for example a query to the data service. Typically, in caBIG®, a data service presents an object view of a data resource that is defined by an information model and typically exposes a well defined query method. For the grid, this consists of a CQL query as an input and CQL result as an output. Whereas in the past, compatibility required linkage of 4 artifacts (An-API (1) , represented by an UML model (2) , -An annotated UML model with controlled vocabulary (3) with the Annotated UML model and vocabulary transformed into metadata (4) (CDEs in caDSR). These services were evaluated by the compatibility checklists and reviews. The current compatibility guidelines would lie in the platform specific model level (PSM) and comprise parts of the information viewpoint, computational viewpoint, and engineering viewpoint.

Note: Section 3.3 is extracted from -SAIF Interoperability Reviews -White Paper; background; https://wiki.nci.nih.gov/x/TJuPAQ. Please refer this source for more details.

3.4 Where do semantics fit in ECCF

Metadata and semantic interoperability will play a very important role in the Informational Viewpoint and computational viewpoint in ECCF. A question remains whether CDEs as caBIG® currently defines them will still play a role.

In trying to answer this question, we realized that there are many unknowns about the specifics of how ECCF will be implemented. Questions such as what repositories will be used to store semantic information, how will these repositories be formulated (ISO 11179 or other), how will searches be run in the caGrid, and more, are still being determined.

We do know that ECCF describes a Specification Stack whose artifacts will provide working interoperability. We know that the artifacts required by the Specification Stack will include semantic profiles. We know that there will be three levels of model abstractions, the CIM, PIM and PSM that can be represented as UML models. Given this information, we can make some generalizations about what semantic information will be important to contribute to working interoperability. We cannot elaborate on specifics of exact levels of semantic detail, such as how semantic information will be stored, how it will be searched, etc.

3.4.1 CDEs and ECCF

In our current caBIG® Silver Compatibility Process, CDEs are defined by the ISO 11179 standard. Going forward into ECCF, it is not clear that the ISO 11179 standard will still be required by all projects. Therefore, we have decided to break down the CDE into its essential parts. What is absolutely necessary for human comprehension of a model? What is absolutely necessary to provide computable semantics - e.g., searching abilities based on the meaning of the components of a model?

For human comprehension of a model, there are two requirements, each class and attribute in that model must have a definition, and each attribute must have an explanation of the how that piece of data will be represented. The definition can simply be text. The representation of data can be satisfied with the ISO 21090 datatypes that have been adopted by NCI CBIIT and

caBIG®. Using these data types will provide a standard, platform-independent description of how data in a project is represented.

A text definition and ISO 21090 data types provide the minimum requirements for human understanding of models in all levels of abstraction, CIM, PIM and PSM. In addition, we believe there is value in each model specifying one or more controlled vocabularies that contain terminology relevant to the domain of the model. These controlled vocabularies can be used to map concepts to the classes and attributes in a model to provide a more rigorous definition and to allow automated searching capabilities. It will be very important that all data elements that are traceable to elements in the caBIG® DAMs (e.g., BRIDG and Life Sciences DAM (LS DAM)) use controlled vocabularies. For other domains, their use is also recommended but not required.

These semantic basics - a text definition, an ISO 21090 representation and a controlled vocabulary - will be useful at all levels of model abstraction, CIM, PIM and PSM. Will CDEs also be needed? We believe at the CIM and probably at the PIM level, creating CDEs as described by ISO 11179 will not be necessary and could, in fact, restrict reuse of models at this abstract level by early binding of class and attribute concepts into a DEC (Data Element Concept). For example, investigating interoperability points at the attribute level rather than the DEC level would allow Patient.Address and Person.Address to map. (Note that it is the early binding of class and attributes that restricts reuse and not early binding to vocabularies.) At the PSM level, mapping CDEs may make more sense. We believe that some projects will want to map their PSM to CDEs while other projects may not have a use case for it. For example, a project that wants to map their model to CRFs may find it useful to map the model to the CDEs created as standards for Case Report Forms (CRF). Also, the linkage of the DEC and value domain into a CDE provides context beyond the concepts and is useful in some instances.

We believe that by mapping concepts from controlled vocabularies to the classes and attributes and by adding ISO data types to the attributes, projects will have the freedom to decide what further semantic artifacts they need. Concept mapping and representation are the basic building blocks of an ISO 11179 Data Element (CDE). As long as model contains these basic building blocks, then each project can decide if they want to create CDEs and store them in an ISO 11179 repository, or not.

4 Recommendations

The “Effect of SAIF-ECCF adoption on CDE reviews” group examined the role of CDE or Data Elements in current semantic and syntactic interoperability and suggests the following semantic requirements for each “cell” on the ECCF specification stack. In addition, the group provides the recommendations for the review process.

4.1 Recommendations for Semantic Requirements

Recommendations for semantic requirements are provided for the Information and Computational viewpoints in the ECCF Specification Stack. We did not identify any semantic

requirement recommendations for the Enterprise, Business and Engineer viewpoints at this point. Specific recommendations are given to each of the three levels of abstraction based on the OMG's Model-Driven Architecture (MDA).

4.1.1 Computationally Independent Model (CIM)

4.1.1.1 Information Viewpoint

At the CIM level, each project needs to provide a Project Analysis Model. This model should originate from caBIG® recognized DAM or DAMs. At DAM level, all classes and attributes have to have clear semantic definitions. That will lead to the most efficient semantic interoperability. In other words, project teams will be able to adopt/adapt the most appropriate class or classes from DAM to Project Analysis Models. Using the same DAM, multiple projects could develop many different Project Analysis Models. It would be highly desirable to have the DAMs available as an EA or ARGO UML file that can be leveraged by the project team for creating their Project Analysis Model. It is recommended that the DAMs be available in a knowledge repository so that project teams could search for the classes and attributes of their interest.

For the Project Analysis Model prepared under the CIM level, the following artifacts are recommended:

- Class and attribute definitions should be made available
- Attribute datatypes are specified as ISO21090
- Concepts are assigned from a controlled vocabulary (as in DAM)
- Vocabulary/vocabularies used are stated

The creation of ISO11179 defined CDEs is not required at this level due to the fact that most users will only focus on the concepts of the classes and their attributes and the generic representation of the attributes. The value domains cannot be made explicit yet, and the creation of a CDE would bind a Data Element Concept (DEC) to a specific value domain, imposing unnecessary constraints on data re-usability. As the assumption is that the DAM(s) will be available in a Knowledge Repository and all the classes and attributes along with their definitions and concepts mappings are borrowed from DAM, the registration of Project Analysis Model would be an open question.

4.1.1.2 Computational Viewpoint

The computational viewpoint is one of the five viewpoints in the RM-ODP reference model for the specification of system architecture for distributed processing of objects. It enables distribution through functional decomposition of the system into objects which interact at interfaces. It describes the functionality provided by the system and its functional decomposition. The artifacts in the ECCF specification stack that are required at the CIM level for the Computational Viewpoint are those for Service Inventory, Service Interface CFSS, and Services roles and relationships.

For the artifacts prepared under the CIM level, the recommendations are-

- Class and attribute definitions should be made available

At this level the focus is on description of the services in order to elucidate the role of each service and relationship between them. Therefore, attribute datatype, concepts, vocabularies, and CDEs would not be required. Class and attribute definitions would be sufficient to provide enough information for understanding the contents of the artifacts that are created at the CIM level for the computational viewpoint.

4.1.2 Platform-Independent Model (PIM)

4.1.2.1 Information Viewpoint

For the PIM level or the logical level, each project will develop a Platform-Independent Model (PIM). This model will be based on the classes and attributes available in the Project Analysis Model prepared for CIM level. At the PIM level, the classes and attributes could be further localized and/or constrained. Using the same Project Analysis Model multiple projects could develop many different PIMs.

For the Project Analysis Model prepared under the PIM level artifacts, the recommendations are divided into two parts i.e. a) classes and attributes that are traceable to a Project Analysis Model and therefore to the DAM and b) the Classes and attributes that representing new content area.

(a) Classes and attributes traceable to a Project Analysis Model and DAM.

These are the classes and attributes that are available in the Project Analysis Model (CIM level) or the new classes and attributes that are introduced in the PIM as a result of the required localizations or constraints. The recommendations for these classes and attributes in the PIM are-

- Class and attribute definitions should be made available
- Attribute datatypes are specified as ISO21090
- Concepts are assigned from a controlled vocabulary as in CIM and can be traced back to DAM where ever possible (existing DAM classes)
- Vocabulary/vocabularies used are stated

The creation of CDE would not be required at this level. We can judge conformance if we register the description of the constraints and the extensions, localizations etc. This can be achieved by making publicly available the UML model, associated class attribute definitions and concepts from a standard vocabulary.

(b) Classes and attribute representing new content area

These classes represent a domain very specific to the project where the developers do not want to bring it to DAM. The recommendations for these classes and attributes include-

- Minimum recommendation is a good textual definition
- Concept mapping is recommended but not required

The creation of Common Data Elements (CDE) would not be required at this level. Registration in a knowledge management system may not be required but there should be a place where projects could register their content if they wish to.

4.1.2.2 Computational Viewpoint

The artifacts that are required at the PIM level for the Computational Viewpoint are those for Service interface PIM and Collaboration Specification. At this level the specification of the services becomes more rigorous and requires a formal model of the service interface and the collaborations. Therefore, the semantic description of the service interface and collaborations become important at this level.

For the artifacts prepared under the CIM level, the recommendations are-

- Class and attribute definitions should be made available
- Attribute datatypes are specified as ISO21090
- Concepts are assigned from a controlled vocabulary
- Vocabulary/vocabularies used are stated

Since the level of specification is still at a logical level and there is not physical interactions between the decomposed objects, CDEs would not be required at this level. The above mentioned semantics would be sufficient to explicitly document the interaction between the objects.

4.1.3 Platform-Specific Model (PSM)

4.1.3.1 Information Viewpoint

For the PSM level or the Implementable level, each project will develop a Platform-Specific Model (PSM). This model will be based on the classes and attributes available in the PIM. However at the PSM level, classes and attributes could be further localized and/or constrained and platform specific bindings and mappings are provided. Using the same PIM, multiple projects could develop many different PSMs.

For the Project Analysis Model prepared under the PIM level artifacts, the recommendations are divided into two parts i.e. a) classes and attributes that are traceable to a Project Analysis Model and therefore to the DAM and b) the Classes and attributes that representing new content area.

(a) Classes and attributes traceable to a PIM-Project Analysis Model (CIM level) and DAM.

These are the classes and attributes that are available in the PIM or the new classes and attributes that are introduced in the PSM as a result of the required localizations or constrained. The recommendations for these classes and attributes in the PSM are-

- Class and attribute definitions should be made available.
- Attribute datatypes specified as ISO21090
- Concepts assigned from a controlled vocabulary as in PIM and can be traced back to PIM, CIM and DAM whereever possible (existing DAM/PIM classes)
- Vocabulary/vocabularies used for assigning concepts are stated.
- Specific permissible values are made available
- Permissible values should have good textual definition
- For Permissible values, concept mapping from a controlled vocabulary is provided.
- For Permissible values, controlled vocabulary is specified

The creation of Common Data Elements (CDE) by registration in caDSR would be an open question at this level. Registration at a metadata repository or some other knowledge management System is strongly recommended.

(b) Classes and attribute representing new content area

These classes represent a domain very specific to the project where the developers do not want to bring it to DAM. The recommendations for these classes, attributes and permissible values include-

- Minimum recommendation is a good textual definition
- Concept mapping is recommended but not required

The creation of Common Data Elements (CDE) by registration in caDSR would be an open question. Registration in a knowledge management system may not be required but there should be a place where projects could register their content if they wish to.

4.1.3.2 Computational Viewpoint

The artifacts that are required at the PSM level for the Computational Viewpoint are those for Interface PSM, Service Realization Specification, and Orchestration Scripts. At this level the specification of the services are the most stringent because technology solutions are realized from this level of specification, so working interoperability between SAIF-ECCF compliant applications are expected to be realized at this level. This level requires a formal model of the service interface, service realization and the orchestrations of the services. Therefore, the semantic description of the service interface and workflow become important at this level.

For the artifacts prepared under the PSM level artifacts the recommendations are-

- Class and attribute definitions should be made available
- Attribute datatypes specified as ISO21090
- Concepts assigned from a controlled vocabulary.
- Vocabulary/vocabularies used for assigning concepts are stated.
- Specific permissible values are made available
- Permissible values should have good textual definition
- For Permissible values, concept mapping from a controlled vocabulary is provided.
- For Permissible values, controlled vocabulary is specified

These semantics would enable working interoperability between objects at the implementation level. Registration in a metadata repository or some other knowledge management system is strongly recommended, since it would facilitate interoperability even further. But the creation of Common Data Elements (CDE) by registration in caDSR would be an open question at this level since an alternate metadata registry could work equally well or possibly better depending on the implementation of caBIG® v2.0.

4.2 Recommendations for Reviews

4.2.1 Current Reviews

The current CDE reviews as part of the Silver Level Compatibility review will unlikely be required as we move forward. Review of CDE standards remain an open question depending on the needs of case report form (CRF) users, the support provided by caBIG® to the case report form user community, and the identification of the best metadata repository or the knowledge management system to support the requirements of caBIG® v2.0. The identification of the best metadata repository for caBIG® v2.0 is left as an open question and can possibly be performed by another group based on the available use cases in the near future.

4.2.2 Semantic Reviews in context of ECCF

Going forward all projects will be based on Domain Analysis Models (DAMs). These DAMs are governed by Composite Architecture Teams (CATs). As mentioned earlier, projects will first build a Project Analysis Model (CIM level) that will contain a subset of DAM classes and attributes that are relevant to the project domain. Next, the project will create a PIM, which is a logical model, based on the Project Analysis Model's classes and attributes, possibly with the addition of other classes and attributes relevant to the project domain that are not part of the DAM. Finally, the project will create the PSM, which is an implementation model based on the PIM, possibly adding or constraining classes and attributes which will be required for the platform specific implementation.

A DAM will contain definitions, ISO 21090 datatype representations and concept mappings that need to be "vetted" by a group of Subject Matter Experts (SMEs). For example, in Life Sciences, the Life Sciences-CAT (LS-CAT) governs the maintenance of the LS-DAM and the content of DAM is reviewed by a group of relevant SMEs. The SMEs include members of Information Representation Working Group (IRWG) within the Integrative Cancer Research (ICR) group. When a project creates a Project Analysis Model (CIM level) by taking the relevant classes and attributes from the DAM, it automatically gets the definitions, concept mappings and ISO21090 representations from the DAM. This ensures reuse because all Project Analysis Model that use that DAM will inherit and use the same semantics to describe the classes and attributes.

The same is true for PIMs and PSMs. Any classes and attributes taken directly from the Project Analysis Model (CIM level) will ensure reuse - their semantics will come from the DAM and so will be the same. At the PIM and PSM levels, there may be new classes and attributes added to the model either as a result of localizations and constraints required by the project or to classes and attributes added to represent the new domain.

Who is responsible for the semantics of these new model components? The project would have to be responsible to provide definitions, representations, and the concept mappings (as appropriate based on the recommendations above). After all, these new classes and attributes were created by the project team, so they are the experts.

But who is the governing body to review and approve the semantics of these new classes and attributes? The VCDE Workspace could play an important role in reviewing and approving

these new semantics while performing the Specification Reviews. The reviews can be conducted in much the same way as Silver Compatibility Reviews were performed previously. The review would be much broader, no longer reviewing CDEs, but now reviewing text definitions, ISO 21090 representations and concept-mappings as appropriate. The alignment between DAMs to Project Analysis Model (CIM level), Project Analysis Model (CIM level) to PIM and PIM to PSM would have to be reviewed as well. It will be important to ensure the transformation between these models as part of the review process, ensuring that the transformation definitions, if required, are provided and are correct. Any new concepts that are required by the project teams would have to be registered in a controlled vocabulary and verified. The project team would be responsible for getting the new terminology registered in the vocabulary and the VCDE WS reviewers can confirm the definition and registration of the new terminology during the Specification Reviews.

There is one more place where reviewing semantics will be important. As more and more projects implement their services using the ECCF, it will become apparent that some classes and attributes created by projects at the PIM (and possibly even the PSM) level are of general interest and should be included in the DAM. Here again, the CAT responsible for that DAM will be the governing body, deciding which classes or attributes are added to the DAM, and deciding on the final official definition and ISO-21090 representation.

The VCDE Workspace can also play an advisory role to the CATs when adding new classes and attributes to DAMs. The VCDE reviewers that review new class and attribute semantics in PIMs and PSMs should be aware of those classes/attributes that are good candidates for being added to the DAM. They could be responsible for harmonizing the semantics of these classes/attributes from the different projects that use them. By doing some up-front harmonization, they can then present a well-thought out class/attribute to the CAT for inclusion into the DAM.

4.2.3 Roles of the VCDE in review of Semantics in ECCF

VCDE Workspace could play a significant role in reviewing and guiding the semantic requirements of ECCF to achieve working interoperability. We see the VCDE being involved in these ways:

- VCDE can perform formal reviews of the DAMs recognized by caBIG®
- VCDE members can help in Specification Reviews by checking the alignment between DAMs to Project Analysis Model (CIM level) to PIM and PSM.
- VCDE can review any new subject area that is presented in the PIM and PSM of development projects.
- VCDE can act as advisors to CAT on recommending new content for inclusion in DAM based on their review and usage (popularity).
- The DAM team, after being advised by the CAT, should communicate with VCDE and the relevant development teams about changes to the DAM.

4.3 Additional Recommendation

Creation of standard models for the CIM level, PIM level and PSM level from DAMs would be desirable since developers could use these models or parts of these models if they wish. This will not only speed up the development process but also help in the alignment reviews (DAM-CIM-PIM-PSM) performed as part of Specification Reviews under SAIF-ECCF implementation at caBIG®.

5 Conclusion

While the specifics of how metadata will be considered in the SAIF/ECCF environment will differ from the past reliance on reuse of CDEs and Silver Level Reviews, many procedures will still be important for ensuring interoperability. In particular, reliance on controlled vocabularies and clear definitions, as well as governance rules will be important to future use cases.

6 References

1. caBIO ECCF wiki (https://wiki.nci.nih.gov/display/caBIO/caBIO+ECCF)

2. caBIO ECCF Artifacts (https://gforge.nci.nih.gov/svnroot/cabiodb/ECCF/artifacts/)

3. Cancer Data Standards Registry (caDSR) Release 4.0 Technical Guide (https://gforge.nci.nih.gov/docman/view.php/59/15452/caDSR_v40_TechnicalGuide.pdf)

4. caGrid 1.3 Document (https://cabig.nci.nih.gov/workspaces/Architecture/caGrid/)

5. Current caGrid Platform - Gap Analysis and Nominal Architecture Draft (https://wiki.nci.nih.gov/x/tSdLAQ)

6. HL7 SAIF Document or Book (http://gforge.hl7.org/gf/download/docmanfileversion/5569/7139/saif_introduction_for_peer_review_20100409.pdf)

7. HL7 ECCF Document (http://gforge.hl7.org/gf/download/docmanfileversion/5524/7020/Enterprise_Conformance_Compliance_Framework_peer_review_20100312.pdf)

8. International Standard ISO/IEC 11179-1, second edition 2004-09-15 (https://www.nescent.org/wg/dryad/images/c/c5/ISO_IEC_11179-1_2004(E)_Metadata_Registries_Framework.pdf)

9. NES Service Specification (https://wiki.nci.nih.gov/display/EAWiki/NES+Specification+Process)

10. SAIF/ECCF training material (https://cabig-kc.nci.nih.gov/Vocab/KC/index.php/ECCF_Training_Feb_2010)

11. SAIF effects on XC WS whitepaper (https://wiki.nci.nih.gov/x/WgNlAQ)

12. SAIF Interoperability Reviews - White Paper; background (https://wiki.nci.nih.gov/x/TJuPAQ)

7 Appendix

7.1 Metadata in caDSR

It is important to understand that what is stored in the caDSR is metadata, not a UML model. In caBIG®, we often refer to "loading models," but what we are really loading to caDSR is the metadata that describes the data represented by that model. That metadata is stored in the caDSR as components, specified by the ISO 11179 standard. The importance of the ISO 11179 standard is to define a framework and required protocols to specify this metadata. The ISO standard ensures that metadata is consistently maintained and sharable across diverse domains. Using the ISO 11179 as a framework is part of what makes caBIG domain models on the Grid semantically interoperable.

The components stored in the caDSR include Object Classes (OCs) and Properties which together make up the Data Element Concepts (DEC). The DEC is a "concept that can be represented in the form of a data element, described independently of any particular representation" (International Standard ISO/IEC 11179-1, second edition 2004-09-15). Also stored are Permissible Values and Value Meanings that make up the list of possible values in a Value Domain. Finally, the DEC and Value Domain together make up the Data Element.

7.1.1 Metadata Components

7.1.1.1 Data Element

A Data Element is composed of the ISO 11179 standard components Data Element Concept (DEC) and Value Domain (VD). The DEC unambiguously describes what a Data Element is. The VD describes the representation of the Data Element.

ISO 11179 Standard definitions-

Data Element Concept: "concept that can be represented in the form of a data element, described independently of any particular representation" (International Standard ISO/IEC 11179-1, second edition 2004-09-15)

Value Domain: "set of permissible values" (International Standard ISO/IEC 11179-1, second edition 2004-09-15)

7.1.1.2 Data Element Concept

A Data Element Concept is composed of the ISO 11179 standard components Object Class (OC) and Property. The Object Class describes a real-world item or concept that has attributes. The Property describes one of those attributes.

ISO 11179 Standard definitions:

Object Class: "set of ideas, abstractions, or things in the real world that are identified with explicit boundaries and meaning and whose properties and behavior follow the same rules" (International Standard ISO/IEC 11179-1, second edition 2004-09-15)

Property: "characteristic common to all members of an object class" (International Standard ISO/IEC 11179-1, second edition 2004-09-15)

7.1.1.3 Value Domain

A Value Domain (VD) describes how a piece of data is represented. It includes information about the data type as well as restrictions on the data values that are permitted for the DE. Value Domains can be non-enumerated, meaning that the permissible values for this DE are specified by a description of some kind, or enumerated, meaning that an explicit list of permissible values are included in the repository as part of that VD.

ISO 11179 Standard definitions:

"A permissible value is a combination of some value and the meaning for that value. The associated meaning is called the value meaning. A value domain is the set of valid values for one or more data elements. It is used for validation of data in information systems and in data exchange. It is also an integral part of the metadata needed to describe a data element. In particular, a value domain is a guide to the content, form, and structure of the data represented by a data element." (International Standard ISO/IEC 11179-1, second edition 2004-09-15)

7.1.1.4 Concept Class

In ISO 11179, a Concept Class is the component that holds information about concepts from vocabularies. Concepts from standard vocabularies serve as the basic building blocks of a Data Element and the components that make up that DE. The current process that uploads metadata into the caDSR requires these concepts to come from the NCI Thesaurus (NCIt). This is a standard vocabulary maintained by NCI-CBIIT. However, the ISO 11179 standard places no restrictions on which vocabularies can be used to create Concept Classes.

Each Object Class and Property in the caDSR has associated with it a series of one or more concepts from the NCIt to completely and unambiguously describe the meaning of that class or attribute. The order of the concepts is very important and contributes to the semantics. The last concept in the series is called the Primary concept and is the base concept for the component. Other concepts qualify the meaning of that Primary concept and are called Qualifier concepts. Even the order of the qualifier concepts can be of significant semantic importance.

The value meanings of permissible values in a Value Domain also can have Concept Classes associated with them. Just like for Object Classes and Properties, the order of these concepts determines the final semantics and unambiguously define the meaning of a permissible value.

7.1.2 Required Data Element ISO 11179 attributes in caDSR

Metadata is currently uploaded to the caDSR by the UML Loader tool. The current UML Loader requires certain attributes described in the ISO 11179 standard to be specified in order to create components for storage in the caDSR. The major components that make up a Data Element include:

- Data Element Concept
- Object Class

- Inheritance
- Association
- Property
- Concept Class
- Value Domain
- Permissible Value
- Value Meaning

The following lists describe the information currently required by the caDSR. Some components require all of this information, other components only a subset of the information.

Information internally created by the UML Loader:

- Public Id
- Version
- Workflow status
- Short name
- Conceptual Domain (for all components except Value Domains)

Information based on attributes from sub-components:

- Long name
- Preferred Definition

Information based on structure of the UML model:

- Classification Scheme Items (UML_PACKAGE_NAME type)
- Alternate Name (UML Qualified Attr type)
- Alternate Name (UML Class:UML Attr type)
- Inheritance of Object Classes
- Associations between Object Classes (includes directionality, multiplicity, role names)
- Value Domain Name
- Permissible Values

Information specified by the model owner in the submission form:

- Context
- Classification Scheme
- Classification Scheme Items (UML_PACKAGE_ALIAS type) (optional)

Information specified by the model owner in the XMI file with tagged values:

- Alternate Definition
- Concept class(es) (includes concept code, name, definition and source)
- Value Domain preferred definition
- Value Domain data type

- Value Domain type (enumerated/non-enumerated)
- Value Domain Conceptual Domain
- Value Domain Representation Term
- Value Meaning (includes concept codes (if applicable), value meaning name and description)

7.2 Specifics on how interoperability is currently achieved on the caGrid

7.2.1 Key technology enablers which allow semantic and syntactic interoperability

- Shared controlled vocabularies, from which "concepts", which are well understood meaning of terms, are derived and used as a means of common understanding of terms between the different data sources and applications.

- Common Data Elements (CDEs), which are data elements that are used in common between the different data sources and applications and which map concepts to actual representations of those concepts in a software system.

- Information Models, which define and encapsulate relationships among CDEs. In general a particular application is represented by an information model which is loaded into the caDSR.

- Application Programming Interface, which are published interfaces of the applications and enable the mechanistic interaction between different applications on the Grid.

7.2.2 Semantic Interoperability

For two systems to be semantically interoperable, they must be able to communicate with each other the meaning of each piece of data in their system. Only then can the two systems know how to use information from the other system. Is a birth date in one system the same as the date of origin in another system? If the two systems use the same Data Element to describe these pieces of information, then you know that they are the same. You know that you are comparing "apples to apples." Further, you know that the two pieces of information are represented identically. This means that one system can directly consume the data of the other system and use it just like it would their own data.

When two systems use the same Data Element, this ensures the highest level of semantic interoperability. But there are other levels of semantic interoperability as well. Perhaps two systems do not share any Data Elements, but they share Object Classes. Now the two systems understand that they both contain the same objects in their domains, but are interested in different attributes of those objects. When two systems share the same DECs, they know that their data have the same meaning, but are somehow represented differently. Perhaps one system represents a birth date as just the month and year, while the other system includes month, year, day and time. Knowing that these two pieces of information use the same DEC to describe their meaning, allows the two systems to know that they have similar information that they may want to compare, but there will need to be some transformation in representation of that data from one system to the other in order to allow either system to consume the data of the other system.

Any ISO 11179 component may be reused by multiple systems. Therefore, two systems can interoperate at any of those levels (DEs, DECs, OCs, Property, VDs, Value Meanings). Whether or not a particular level of interoperability is of importance to a system depends on how that system wants to consume the data in the other system. There may be times when it will be important that two systems share Properties, even when they do not share any DECs. And there will be other (probably much more numerous) times when Property reuse will not be enough information for two systems to interoperate in a meaningful way.

The current process in caBIG® to describe interoperability has been for each system to upload the metadata that describes the data in their system to the caDSR. The NCI Thesaurus is used as the standard vocabulary and the ISO 11179 standard is used to create the components that are uploaded into the caDSR. Systems that use these standards can then be placed on the caGrid and can interoperate with other systems that also use the same standards. Further, systems can also be certified as Silver Compatible by going through a review that looks for semantic interoperability by reviewing the metadata that the system uploaded into caDSR and looking for completeness and reuse of metadata used by other systems.

7.2.3 Syntactic interoperability

Syntactic interoperability is achieved when systems exchange data through shared interfaces. Regularization of application programming and messaging interfaces is necessary to overcome obstacles to syntactic interoperability.

The caBIG® program currently provides tools and processes for creating a semantically and syntactically interoperable environment based on the use of Grid services.

Currently, developers have to meet certain basic requirements in order to deploy a "semantically annotated" Grid service:

- Concepts representing the domain must exist in a terminology server (EVS).
- Common Data Elements (ISO11179), which use those concepts and controlled vocabulary along with other information, must exist for every class and attribute that you plan on using in your object model.
- An object model that has every class and attribute annotated with CDEs must exist which represents the data types you will be using.
- A schema must be generated that reflects how the object model will actually look when serialized to XML.
- The annotated object model must be submitted for review to NCI CBIIT for review and acceptance.
- The annotated model must have a corresponding physical data model that describes exactly which class and attributes go into wich tables and rows.

Once the model is approved, the caCore and caGrid development tools can be used to create and expose the grid service.

caCore SDK generates the data service backend from models that have been approved and uploaded into caDSR/EVS and then updated with schema information. A grid data service can

then be created using the wizards and tooling available through caGrid. The main tool is Introduce with the data service wizard, but it can only easily create a data service that was generated from a caCORE SDK system.

On the caGrid, objects exposed as data services comply with common data elements registered in the caDSR/EVS, and are transported as XML using schema types registered in the GME (Global Metadata Extract). The GME provides run-time-accessible maps via QNames that link to the conceptual-level representation defined and stored in caDSR. The caDSR is intended to ensure semantic interoperability, while the GME is intended to ensure programmatic data exchange (syntactic) interoperability.

CDEs aid in the syntactic interoperability on the caGrid. Elements in caGrid can be aggregated based on components of the CDEs, such as the object class, attribute and associations. In particular, query by example can be used on caGrid using restrictions on these components of CDEs across multiple data services and aggregated sets of data meeting the criteria will be returned.

However, there is currently little ability to interoperate during workflow on caGrid. "There is no way to link a registered model to a run-time schema representation, a fact that directly impedes predictable Computable Semantic Interoperability (CSI). If the infrastructure were truly dependent on the semantics expressed by ISO 11179-compliant metadata environment, the registration processes would need to define both the data's semantics and its representation on the wire (e.g. via binding of the data element to a data type standard such as ISO 21090). At present, only a conceptual-level representation is defined and stored in caDSR with multiple run-time-accessible maps via QNames in the Global Metadata Extract (GME)." (https://wiki.nci.nih.gov/x/tSdLAQ)